IEEE Open Journal of the Communications Society

Received 18 August 2022; revised 16 September 2022; accepted 17 September 2022. Date of publication 21 September 2022; date of current version 3 October 2022.

Digital Object Identifier 10.1109/OJCOMS.2022.3208323

Machine Learning for CSI Recreation in the Digital Twin Based on Prior Knowledge

BRENDA VILAS BOAS [1,2] (Graduate Student Member, IEEE), WOLFGANG ZIRWAS [1] (Senior Member, IEEE), AND MARTIN HAARDT [2] (Fellow, IEEE)

[1] Nokia Standards, Nokia, 81541 Munich, Germany

[2] Communications Research Laboratory, Ilmenau University of Technology, 98693 Ilmenau, Germany

CORRESPONDING AUTHOR: B. VILAS BOAS (e-mail: brenda.vilas_boas@nokia.com)

This work was supported in part by the German Ministry of Education and Research (BMBF) under Grant 16KIS1184 (FunKI).

ABSTRACT Knowledge of channel state information (CSI) is fundamental to many functionalities for mobile communication systems. With the advance of machine learning (ML) and digital maps, i.e., digital twins, we have a big opportunity to learn the propagation environment and design novel methods to derive and report CSI. In this work, we propose to combine untrained neural networks (UNNs) and conditional generative adversarial networks (cGANs) for MIMO channel recreation based on prior knowledge. The UNNs learn the prior-CSI for some locations which are used to build the input to a cGAN. Based on the prior-CSI estimates, their locations and the location of the desired channel, the cGAN is trained to output the channel expected at the desired location. This combined approach can be used for low overhead CSI reporting as, after training, we only need to report the desired location. Our results show that our CSI recreation method is successful in modelling the wireless channel under different configurations of prior-CSI spatial sampling. In addition, the results consider a real world measurement campaign for indoor line of sight and non-line of sight channels. The signal to noise ratio (SNR) achieved by our CSI recreation is better than the SNR reported by the measured campaign providers. Moreover, our CSI recreation provides means for low overhead CSI reporting as the UNN structure is underparameterized compared to the full explicit CSI, and only the desired location is needed for the cGAN to recreate the desired CSI.

INDEX TERMS Channel estimation, channel interpolation, UNN, cGAN, digital twin.

I. INTRODUCTION

VISIONS for the 6th generation of mobile communications point towards the fusion of the real and the digital worlds, e.g., mixed-reality experiences, where the network needs to develop useful knowledge of the physical world based on collected data [1]. In that context, artificial intelligence and machine learning (AI/ML) applications play an important role for the design of 6G systems. Today, AI/ML applications for physical layer are gaining momentum in standardization bodies, such as 3GPP [2] and O-RAN [3]. Specifically for 3GPP release 18, a study item was initiated in the physical layer working group (RAN 1) to discuss AI/ML for the new radio (NR) air interface [4].

A digital twin is a virtual model that represents at least in some relevant and predefined aspects a real object. The digital model conveys real measurements, e.g., from sensors, and simulations which allow to generate insights about the physical object [5]. The concept arose from the product life-cycle management area within NASA where different sensed data would be conveyed in the digital twin to continuously monitor the health of the system, i.e., a flying vehicle [6]. From a wireless communications prospective, combining AI/ML capabilities with virtual representations of the real world, enables a variety of possibilities for wireless network planning, deployment, and management. For the radio access network (RAN), the digital twin could run at the base station (BS) side processing collected data to store environmental characteristics. For instance, the collected data can be user equipment (UE) measurements, LIDAR data, or ray-tracing simulations. This aggregated memory is used in our case as prior knowledge to interpolate the channel state information (CSI) and to reduce the reporting overhead. In order to leverage the potential of a digital twin for the wireless propagation environment, full knowledge of the CSI for a set of known locations is desired such that most of the real propagation effects can be represented. In this work, CSI recreation refers to the combination of channel estimation and channel interpolation for the purpose of reducing the CSI reporting overhead. We propose to combine two AI/ML methods, untrained neural networks (UNNs) and a conditional generative adversarial network (cGAN), for channel recreation on the digital twin. The main goal is to minimize the overall complexity of the neural networks (NNs), reduce the training time, and enable low CSI reporting overhead. In that context, the UNNs provide accurate CSI for some known locations, while the cGAN provides some interpolation functionality to minimize the required number of locations with accurate CSI.

The works in [7], [8], [9], and [10] have shown how to combine multi-modal data from ray-tracing simulations, lidar, and environment images for beam selection, either in vehicular scenarios or in scenarios with unmanned aerial vehicles. In all those papers, the CSI is generated from ray-tracing and constitutes a feature for deciding on the best beam. Furthermore, [10] shows an idea of what a digital twin of the wireless propagation scenario should look like. The authors in [11] propose to estimate large scale fading maps based on top-view images of the propagation environment. Their FadeNet method is inspired by the U-Net NN architecture, and it is trained under supervised learning using ray-tracing simulations as ground truth. In contrast to the state of the art, our solution does not rely on multi-modal data, such as lidar or environment images, which allow us to reduce the complexity of our NN architectures. Our CSI recreation method is different from RF fingerprinting algorithms mainly because we are interested in the small-scaling fading characteristics of the propagation environment, and RF fingerprinting usually relies on the received signal strength [12], [13].

UNNs were first proposed in [14] to solve inverse problems, such as denoising, super-resolution and inpainting. The term 'untrained' refers to an algorithm that avoids a huge data collection phase as the updates of the gradient descent are for a single image measurement. The *deep decoder* architecture as proposed in [15] simplifies the structure of a UNN, making it underparameterized. For wireless communications, this means that we can fit a UNN to directly estimate the wireless channel based on a small noisy measurement campaign, i.e., a few time snapshots, without the need of noiseless labels. The work in [16] has proposed the use of UNNs for MIMO channel estimation under pilot contamination. Despite the limitation to statistical channel models, UNNs could reduce the noise level of the measured signal. In our recent work [17], we have shown that UNNs store prior knowledge on the propagation environment. Our UNN estimator provided an estimation gain of about 10 dB in the low SNR regime when compared with the minimum mean squared error (MMSE) estimator. Moreover, the stored prior knowledge can be accessed by transfer learning, which improves the MIMO channel estimation performance. Our current work differentiates from [16] in the usage of the UNN algorithm and its architecture. The novelty of our work consists in using the UNN structures for low overhead CSI reporting and taking their CSI estimates as prior information for the cGAN.

The simplicity of UNNs comes at the cost of a lack of generalization. Since there is no dataset collection for the updates of the weights update, iterating the gradient descent algorithm is always needed when a new set of channel measurements is acquired [17]. Different from UNNs, generative adversarial networks (GANs) need a big data collection and training phases, which steers the generator neural network to mimic the data distribution [18] and leads to a high generalization capability. For wireless mobile radio systems, GANs are often concerned with physical layer issues such as channel modeling and data augmentation [19], [20]. Recently, [21] has proposed to use a Wasserstein GAN to estimate the wireless channel and later adjust the GAN's input random vector to improve the channel estimates. In [22], the authors study the GAN for wideband channel estimation. Conditional GANs (cGANs) provide some prior-knowledge to their generator and discriminator NNs which should ease the mapping task [23]. A cGAN is used in [24] to estimate the millimeter wave (mm-Wave) *virtual* channel covariance matrix based on prior knowledge of a training sequence. A cGAN and a variational autoencoder (VAE) GAN are used in [25], [26], but in a context of end-to-end learning where the final objective is to predict the transmitted symbols, not the wireless channel. In our recent work [27], we have proposed to use a cGAN for channel estimation in MIMO arrays with mixed radio frequency chains, where in a part of the array, antenna elements were turned-off. Our results demonstrated the good generalization capability of cGANs. Our current work differentiates from applications of GANs, such as in [21], because the input to the algorithm is random noise, while in cGANs the input contains information with physical meaning to the problem, e.g., CSI of neighbouring UEs. The use case of cGANs proposed here is completely different from [27], where the missing parts, aimed to be estimated, were the channels of some antenna elements within a single time snapshot MIMO orthogonal frequency-division multiplexing (OFDM) radio channel.

Motivated by the generalization capabilities of cGANs and the underparameterization of UNNs, we propose to combine them for MIMO channel recreation within a digital twin of a propagation area. The main contributions of this paper are:

- We introduce a new CSI recreation method where, first, the UNNs are used to generate prior-CSI estimates for a set of locations. Second, the cGAN uses the prior-CSI estimates together with their locations to recreate the CSI in a desired location. Hence, our approach can be used to add the small scale fading characteristics of the wireless channels to the digital twin. To the best of the authors' knowledge, this is the first work proposing to combine UNNs with cGANs for CSI recreation.

- We present how to perform regular grid-like CSI interpolation with UNNs and, based on these results, motivate the choice of cGANs for CSI recreation. This is the first work that shows the applicability of UNNs for the purpose of channel interpolation. Moreover, we compare them with standard interpolation using FIR filters.
- We present results for CSI recreation using real world channel measurements of line of sight (LOS) and non-line of sight (NLOS) indoor propagation environments. Moreover, we evaluate the performance of CSI recreation under different spatial sampling strategies of prior-CSI measurements for the UNN CSI estimators. In addition, we set up experiments to assess the importance of the location matrix to the cGAN models, and the effect of inaccurate location estimates on the CSI recreation performance.
- Moreover, after deriving the weights for all the AI/ML models, only the desired location needs to be reported. Therefore, our solution enables low CSI reporting overhead if the AI/ML models are exchanged between the UEs and the BS.

In this paper, Section II introduces our proposed method, Section III presents details about our UNN for prior knowledge CSI estimation and how to derive a UNN CSI interpolator from a UNN CSI estimator. Next, Section IV shows the processing performed at the cGAN for CSI recreation using location and prior-CSI, Section V presents our simulation methodology and results, and Section VI concludes our paper.

Regarding the notation, a, $\mathbf{a}$, $\mathbf{A}$ and $\boldsymbol{\mathcal{A}}$ represent, respectively, scalars, column vectors, matrices and D-dimensional tensors. The superscript ^{T}, denotes transposition. For a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times M_2 \times \ldots M_D}$, M_d refers to the tensor dimension in the d^{th} mode. A d-mode unfolding of a tensor is written as $[\boldsymbol{\mathcal{A}}]_{(d)} \in \mathbb{C}^{M_d \times M_1 \ldots M_{d-1} M_{d+1} \ldots M_D}$ where all d-mode vectors are aligned as columns of a matrix following a forward cyclical index ordering. The d-mode vectors of $\boldsymbol{\mathcal{A}}$ are obtained by varying the d^{th} index from 1 to M_D and keeping all other indices fixed. Moreover, $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{A}} \times_d \mathbf{U} \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_{d-1} \times J \times M_{d+1} \times \cdots \times M_D}$ is the d-mode product between a D-way tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times M_2 \cdots \times M_D}$ and a matrix $\mathbf{U} \in \mathbb{C}^{J \times M_d}$ [28] In addition, the concatenation operation $[\boldsymbol{\mathcal{A}} \sqcup_d \boldsymbol{\mathcal{B}}]$ denotes the concatenation of $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ along the d^{th} mode. The concatenation $\sqcup_d$ operation also applies to matrices.

II. CSI RECREATION WITH PRIOR KNOWLEDGE

Here, we propose a functionality to the digital twin that we name *CSI recreation*. In the simplest case, the digital twin is a map that we need to populate with channel measurements to allow better performance of other functions, such as beam management, hand over, and network management.

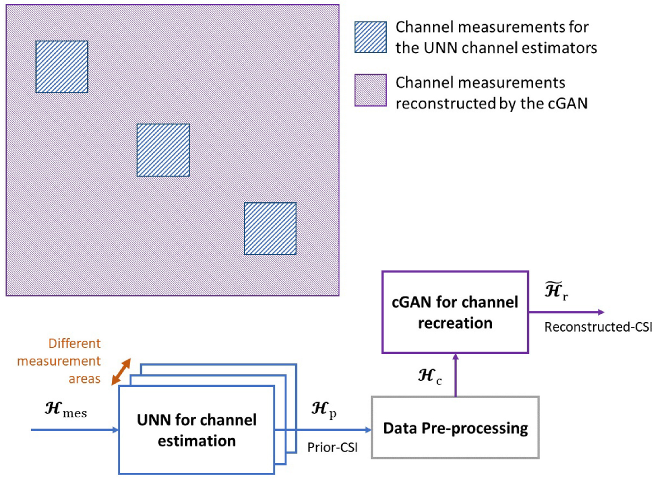

FIGURE 1. Schematic of our proposed ML solution for CSI recreation for a digital twin. Each measurement campaign is represented by a blue box where the UNN performs channel estimation. In purple, we show the second ML part where a cGAN is trained with knowledge of the prior-CSI (in blue), and the location of the desired channels. The cGAN output is the CSI recreation of the desired channels.

The channels that characterize the digital twin can be collected from measurement campaigns or realistic simulations. Depending on the size of the propagation area that we consider, this data collection phase may take a long time, which is not desired. Hence, we propose the CSI recreation, where a neural network (NN) is trained to recreate wireless channels based on the knowledge of a few real channel measurements. More specifically, we combine a UNN for channel estimation of the prior-CSI with a cGAN for the final channel recreation at the desired locations, as illustrated in Figure 1.

In this work, we consider a massive MIMO OFDM wireless channel $\boldsymbol{\mathcal{H}} \in \mathbb{C}^{N_{\text{sp}} \times N_{\text{sub}} \times N_{\text{ant}}}$, at a fixed BS equipped with an uniform rectangular array (URA) containing N_{ant} antenna elements, a single-antenna UE that is moving along a prescribed trajectory, operating with N_{sub} OFDM subcarriers, and collecting N_{sp} time snapshots. Each time snapshot is collected in a Cartesian location point $\boldsymbol{\Gamma} = \{\mathbf{x}, \mathbf{y}, \mathbf{z}\} \in \mathbb{R}^{N_{\text{sp}} \times 3}$ relative to the BS position. The wireless channels for the total area of the digital twin under consideration is referred as $\boldsymbol{\mathcal{H}}$, from which a small part $\boldsymbol{\mathcal{H}}_{\text{mes}}$ is selected for the prior-CSI estimation.

Figure 1 shows our proposed AI/ML framework to recreate CSI in a certain location area (in purple) based on channel measurements of neighboring UEs (in blue). There are two AI/ML instances which collaborate to recreate the CSI at a desired location, the first in blue and the second in purple. The first AI/ML instance aims to find the prior-CSI $\boldsymbol{\mathcal{H}}_{\text{p}} \in \mathbb{C}^{N'_{\text{sp}} \times N_{\text{sub}} \times N_{\text{ant}}}$ based on the measured channels $\boldsymbol{\mathcal{H}}_{\text{mes}} \in \mathbb{C}^{N'_{\text{sp}} \times N_{\text{sub}} \times N_{\text{ant}}}$, where $N'_{\text{sp}} \ll N_{\text{sp}}$. We employ a UNN for this purpose where each UNN estimates $\boldsymbol{\mathcal{H}}_{\text{p}} = [\mathfrak{Re}\{\boldsymbol{\mathcal{H}}_{\text{p}}\} \sqcup_3 \mathfrak{Im}\{\boldsymbol{\mathcal{H}}_{\text{p}}\}] \in \mathbb{R}^{N'_{\text{sp}} \times N_{\text{sub}} \times 2N_{\text{ant}}}$, the channel of a single UE over multiple time snapshots. Even though UNNs are structures with low complexity [15], deriving one UNN model for each possible location in a propagation environment is unfeasible as the weights should be adapted by new gradient iterations [17]. Therefore, we propose to use a second AI/ML instance based on a cGAN for the interpolation functionality due to its generalization capabilities. The second AI/ML instance is trained to compute the recreated-CSI $\mathcal{H}_r \in \mathbb{R}^{(S+1)\times N_{\text{sub}}\times(2N_{\text{ant}}+1)}$ in the desired location $\mathbf{\Gamma}_r \in \mathbb{R}^{1\times 3}$ based on the knowledge of a sub-set of S selected prior-CSI $\mathcal{H}_c \in \mathbb{R}^{(S+1)\times N_{\text{sub}}\times(2N_{\text{ant}}+1)}$ and their respective locations $\mathbf{\Gamma}_p \in \mathbb{R}^{S\times 3}$, more details will be provided in Section IV.

Since UNNs do not need 'labels' to find their best weights, we can perform a small measurement campaign and use the UNN-estimated channels as conditional input to the cGAN. In a 'day-zero' operation where not many CSI measurements are available, the cGAN can be trained with target channels (purple area in Figure 1) derived from simulations. Then, the prior-CSI from the UNNs can help to adjust the model to real propagation conditions. In the long run, we could update the cGAN model based on collected real world measurements. In this scenario, the availability of priors at the conditional input reduces the complexity of the NN structure and its training time if compared to common GANs. In the following sections, we explain in detail how each part of the algorithm is trained.

Due to the low computational complexity of UNNs, the UE can derive the UNN weights and send it to the BS. Then, the BS is able to reconstruct the prior-CSI and can train a cGAN to recreate the CSI at a desired location, which is different from the prior-CSI locations. Since the BS collects and stores all the reported prior-CSI, there is no direct collaboration between the UEs for CSI recreation. After deriving the optimum weights of the cGAN, the UEs within the representation area can report their location $\mathbf{\Gamma}_r$ to the BS instead of the full explicit CSI, which reduces the CSI feedback overhead. Currently, the 3GPP type II CSI reporting is a codebook based feedback that does not allow to recover all the channel coefficients, which limits the use of these reports. Once the CSI recreation is trained and the digital twin is populated with environment knowledge, such as the propagation channel, a map of the scenario, and other side information, there is plenty of information available within the digital twin that can be exploited for CSI prediction or other applications. For improved reliability of the CSI recreation method, the BS can send a trained cGAN model to the UE. Hence, the UE is able to identify when the BS will fail on its CSI recreation and may trigger a correction procedure.

III. ESTIMATION OF PRIOR-CSI WITH UNNS

The underparametrization of a deep decoder [15] and its capability to optimize noisy measurements and recover missing pixels of an image have motivated us to investigate UNNs for prior-CSI estimation and prior-CSI interpolation. Since a UNN does not need the true-labels for computing the gradient iterations, the UE can perform a small measurement campaign and directly use the channel measurements for finding the best UNN weights. Different from prior art, in this section we also describe how UNNs can be used for CSI interpolation. In Section V, we show that this interpolation capability is limited to successive close by measurements. The following sections present the data pre-processing, the UNN architecture and how the gradient descent algorithm is used to update the UNN weights for channel estimation and channel interpolation.

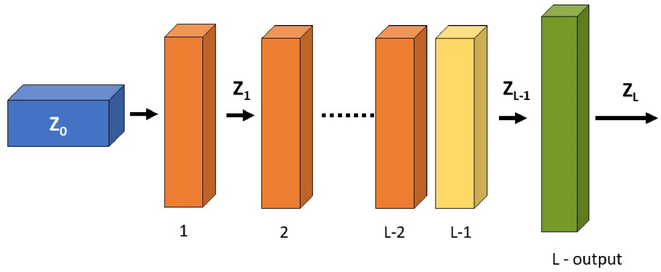

FIGURE 2. General layer structure of a UNN P used to estimate the prior-CSI $\mathcal{H}_p = P(\mathcal{K}^*, \mathcal{Z}_0)$, where $\mathcal{K}^*$ is the collection of optimum weights for the L layers. There are $L-2$ inner layers in orange, one pre-output layer in yellow, and one output layer in olive. In blue, we represent $\mathcal{Z}_0$ the random input tensor.

A. DATA PRE-PROCESSING FOR UNN

The input signal to a UNN is a random noise seed $\mathcal{Z}_0 \in \mathbb{R}^{b\times c\times k_1}$, where $b = N'_{\text{sp}}/2^{L-2}$, $c = N_{\text{sub}}/2^{L-2}$, k_1 is the number of filters in the first layer, and L is the number of layers. The input tensor $\mathcal{Z}_0$ is drawn from a uniform distribution $U(-a, +a)$ defined on the interval $[-a, +a]$ and kept fixed during the iterations to update the gradient descent. The measured channel $\mathcal{H}_{\text{mes}}$ is preprocessed as:

- $\mathcal{H}_{\text{mes}}$ is normalized by its Frobenius norm, which is computed as

$$\|\mathcal{H}_{\text{mes}}\|_F = \sqrt{\sum_{i=1}^{N'_{\text{sp}}}\sum_{j=1}^{N_{\text{sub}}}\sum_{m=1}^{N_{\text{ant}}} |h_{\text{mes}_{i,j,m}}|^2}, \tag{1}$$

  and then multiplied by a scaling factor. This procedure is taken to control the range of the values of the channel coefficients. The values should not be too small and should be within the operational range of the activation function in the output layer. Hence, this preprocessing step helps to ease convergence.
- $\mathcal{H}_{\text{mes}} \in \mathbb{C}^{N'_{\text{sp}}\times N_{\text{sub}}\times N_{\text{ant}}}$ is rearranged by concatenating $[\mathfrak{Re}\{\mathcal{H}_{\text{mes}}\} \sqcup_3 \mathfrak{Im}\{\mathcal{H}_{\text{mes}}\}]$ in the dimension corresponding to the antenna elements.

After those operations, $\mathcal{H}_{\text{mes}} \in \mathbb{R}^{N'_{\text{sp}}\times N_{\text{sub}}\times 2N_{\text{ant}}}$ is directly used to compute the cost function.

B. UNN ARCHITECTURE

A UNN is a composition of L layers where there are $(L-2)$ inner layers, one pre-output layer $(L-1)$ and one output layer (L), according to the deep decoder architecture [15]. Figure 2 shows a generic organization of those layers, the random noise seed $\mathcal{Z}_0$ in blue, the inner layers in orange, the pre-output layer in yellow, and the output layer in olive. All the layer types contain convolutional filters $\mathcal{W}_l \in \mathbb{R}^{1\times 1\times k_{l-1}\times k_l}$ where $l = \{1, 2, \ldots L\}$, k_{l-1} and k_l are hyper-parameters which define the number of filters on the respective $(l-1)^{\text{th}}$ and l^{th} layers. However, the types of layers differ with respect to the upsampling computation and the operation of the batch normalization (BatchNorm) [29]. The inner layers contain linear and non-linear operations. First, there is a convolutional filter $\mathcal{W}_l$ where the weights are updated by using gradient descent. Second, there is a fixed bilinear upsampling operation, where $\mathbf{A}_l \in \mathbb{R}^{2^l b\times 2^{l-1}b}$ and $\mathbf{C}_l \in \mathbb{R}^{2^l c\times 2^{l-1}c}$ are the linear upsampling matrices in the subcarrier and time snapshots dimensions, respectively. Third, the rectifier linear unit (ReLu) activation function is applied, and a batch normalization is computed per k_l filter as

$$\text{BatchNorm}\left(\mathcal{Z}_{lj}\right) = \frac{\mathcal{Z}_{lj} - \text{mean}\left(\mathcal{Z}_{lj}\right)}{\sqrt{\text{var}\left(\mathcal{Z}_{lj}\right)}}\gamma_{lj} + \beta_{lj}, \tag{2}$$

where $j = [1, 2, \ldots, k_l]$, the mean and variance (var) are computed among the batch samples [29], which corresponds to the number of data samples processed before updating the model's internal parameters. The trainable parameters of the BatchNorm operation are $\mathbf{R}_l = [\boldsymbol{\gamma}_l, \boldsymbol{\beta}_l] \in \mathbb{R}^{k_l\times 2}$. The computation performed at each l^{th} inner layer can be written as

$$\mathcal{Z}_l = \text{BatchNorm}\Big(\text{ReLu}\Big(\mathcal{Z}_{l-1} \times_1 \mathbf{A}_l \times_2 \mathbf{C}_l^T \times_3 [\mathcal{W}_l]_{(4)}\Big)\Big), \tag{3}$$

where $[\mathcal{W}_l]_{(4)}$ is the 4-mode unfolding of the convolutional filter operating at the antenna elements dimension. For example, the output of the first inner layer $\mathcal{Z}_1$ can be written as

$$\mathcal{Z}_1 = \text{BatchNorm}\Big(\text{ReLu}\Big(\mathcal{Z}_0 \times_1 \mathbf{A}_1 \times_2 \mathbf{C}_1^T \times_3 [\mathcal{W}_1]_{(4)}\Big)\Big). \tag{4}$$

The pre-output layer $(L-1)$ differs from the inner layers because it does not apply upsampling. Hence, it can be written as

$$\mathcal{Z}_{L-1} = \text{BatchNorm}\Big(\text{ReLu}\Big(\mathcal{Z}_{L-2} \times_3 \big[\mathcal{W}_{L-1}\big]_{(4)}\Big)\Big). \tag{5}$$

Next, the output layer is used to adjust the number of filters of the pre-output layer to the size expected at the output $k_L = 2N_{\text{ant}}$ as

$$\mathcal{Z}_L = \text{TanH}\big(\mathcal{Z}_{L-1} \times_3 [\mathcal{W}_L]_{(4)}\big), \tag{6}$$

where $\mathcal{W}_L \in \mathbb{R}^{1\times 1\times k_{l-1}\times 2N_{\text{ant}}}$, and TanH is the hyperbolic tangent activation function. Since the upsampling operations are pre-defined, the trainable parameters relate to the convolutional filters $\mathcal{W}_l$ and the regularization parameters $\mathbf{R}_l$ of the batch normalization operation. Therefore, $\mathcal{K}_l = \{\mathcal{W}_l, \mathbf{R}_l\}$ is the set of trainable parameters of the l^{th} layer, and $\mathcal{K}$ refers to all trainable parameters of the L layers.

C. UPDATING THE WEIGHTS OF THE UNN CHANNEL ESTIMATOR

Here, we refer to the UNN as a model $P : \mathbb{R}^N \to \mathbb{R}^{N_{\text{sub}}N'_{\text{sp}}2N_{\text{ant}}}$ where $N < N_{\text{sub}}N'_{\text{sp}}2N_{\text{ant}}$ is the total number of parameters. The UNN P performs the mapping operation $\mathcal{Z}_L = P(\mathcal{K}, \mathcal{Z}_0)$, where $\mathcal{Z}_0$ is the random noise seed, and $\mathcal{K}$ is the tensor of weights that represents all the UNN trainable parameters.

The cost function is the mean squared error (MSE), calculated as

$$\mathcal{L}(\mathcal{K}) = \mathbb{E}\Big\{\|P(\mathcal{K}, \mathcal{Z}_0) - \mathcal{H}_{\text{mes}}\|_{\text{F}}^2\Big\}. \tag{7}$$

The gradient descent is updated as in supervised learning, performing I gradient algorithm iterations until the optimum parameters are found, such that

$$\mathcal{K}^* = \underset{\mathcal{K}}{arg\,min}\ \mathcal{L}(\mathcal{K}),\ \text{ and } \mathcal{H}_{\text{p}} = P\big(\mathcal{K}^*, \mathcal{Z}_0\big) \tag{8}$$

is the channel estimation of the prior-CSI. From the loss function, we observe that the prior-CSI $\mathcal{H}_{\text{p}}$ derived by the UNN P is specific to $\mathcal{H}_{\text{mes}}$. Hence, the model P does not directly generalize for other channels, it is specific to the $\mathcal{H}_{\text{mes}}$ considered during gradient updates.

D. UPDATING THE WEIGHTS OF THE UNN CHANNEL INTERPOLATOR

Since UNNs exploit the correlations between successive channel measurements, we show how they can be adapted for CSI interpolation. This operation mode can further reduce the number of prior-CSI measurements needed. As the correlation between the channels reduces with the distance between successive time snapshots, in Section V we show that this approach has limitations which motivates the use of cGANs for CSI recreation.

The same UNN model P can be used for channel interpolation. The difference here is that $\mathcal{H}'_{\text{mes}}$ has a lower resolution when compared to $\mathcal{H}_{\text{mes}}$. This low resolution is represented by a 1/0 mask $\mathcal{M} \in \mathbb{R}^{N'_{\text{sp}}\times N_{\text{sub}}\times N_{\text{ant}}}$, such that

$$\mathcal{H}'_{\text{mes}} = \mathcal{M} \odot \mathcal{H}_{\text{mes}}, \tag{9}$$

where $\odot$ is an element-wise multiplication. Hence, the mask represents time snapshots or subcarriers that are not present in the initial measurement campaign, but we aim to have them at the output of the UNN CSI interpolator. For that purpose, the loss function becomes

$$\mathcal{L}(\mathcal{K}) = \mathbb{E}\Big\{\|\mathcal{M} \odot P(\mathcal{K}, \mathcal{Z}_0) - \mathcal{H}'_{\text{mes}}\|_{\text{F}}^2\Big\}, \tag{10}$$

and Gaussian noise $\mathcal{N}(0, \sigma^2)$ with zero mean and variance σ^2 is added to the input random seed tensor $\mathcal{Z}_0$ every $I_t < I$ iterations as a regularization measure. After I gradient iterations, the prior-CSI from the UNN CSI interpolator is

$$\mathcal{H}_{\text{p}} = P\big(\mathcal{K}^*, \mathcal{Z}_0\big). \tag{11}$$

This procedure is analogous to inpainting for a deep decoder [15], where a UNN outputs missing parts of an image.

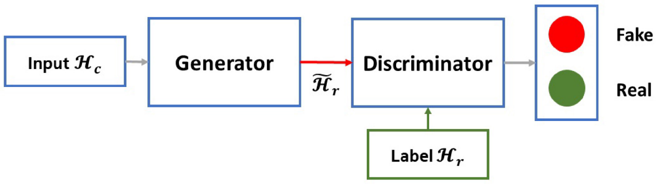

FIGURE 3. Conditional GAN, two NNs play a minmax game where the generator tries to fool the discriminator. The discriminator should classify $\tilde{\mathcal{H}}_r$ as a fake sample, while $\mathcal{H}_r$ is classified as a real sample. The generator fools the discriminator when $\tilde{\mathcal{H}}_r$ is classified as real.

IV. CSI-RECREATION WITH cGAN

UNNs have interesting denoising and interpolation capabilities [14], [15]. However, they use iterative algorithms that have to be adapted for each new set of channel measurements. This additional algorithm latency is not desired. Hence, we propose to combine a few UNNs with a cGAN to recreate the CSI in a digital twin over a larger area. The channels estimated by the UNNs compose the prior knowledge which is given to the cGAN which is composed of a generator NN and a discriminator NN which compete in a minmax game during the training phase, see Figure 3. The generator NN follows an encoder-decoder architecture where the prior-knowledge on the propagation environment is the 'semantic context' [30] that should be leveraged by the encoder to derive latent feature representations which are used by the decoder to produce the desired wireless channel. In this regard, cGANs constrain the solution search space as prior information is available, while in common GANs only the noise is seen at the generator's input. In contrast to UNNs, cGANs require data collection and training to be able to model the data distribution. Nonetheless, it has great generalization capabilities [27]. In the following sections, we present the dataset preprocessing, our cGAN architecture, and the adversarial training.

A. DATASET PREPROCESING FOR cGAN

In this section, we present how we construct the signals to train the cGAN: the conditional input $\mathcal{H}_\mathrm{c}$, the label $\mathcal{H}_\mathrm{r}$, and the generator output $\tilde{\mathcal{H}}_\mathrm{r}$.

The conditional input to our cGAN is derived from the prior-CSI $\mathcal{H}_\mathrm{p}$, their locations $\mathbf{\Gamma}_\mathrm{p}$, and the desired location $\mathbf{\Gamma}_\mathrm{r}$ where we aim to recover the CSI of a certain UE. Each $\mathcal{H}_\mathrm{p} \in \mathbb{R}^{N'_\mathrm{sp}\times N_\mathrm{sub}\times 2N_\mathrm{ant}}$ estimated by a UNN has CSI for N'_sp different locations. Therefore, if N_UE UNNs are used to estimate the prior-CSI $\mathcal{H}_\mathrm{p}^{N_\mathrm{UE}} \in \mathbb{R}^{N_\mathrm{UE}N'_\mathrm{sp}\times N_\mathrm{sub}\times 2N_\mathrm{ant}}$, there are $N_\mathrm{UE}N'_\mathrm{sp}$ CSI-location pairs $\{\mathbf{H}_{\mathrm{p}j}, \mathbf{\Gamma}_{\mathrm{p}j}\}$ available, where $\mathbf{H}_{\mathrm{p}j} = \mathcal{H}_\mathrm{p}^{N_\mathrm{UE}}(j,:,:)$ and $j = \{1, 2, \ldots N_\mathrm{UE}N'_\mathrm{sp}\}$. From the available CSI-location pairs, a sub-set of S CSI-location pairs is selected according to their minimum Euclidean distance to $\mathbf{\Gamma}_\mathrm{r}$, as we assume that the closer the channels are, the higher their chance of being correlated. The S selected prior-CSI positions $\mathcal{H}_\mathrm{p}^S \in \mathbb{R}^{S\times N_\mathrm{sub}\times 2N_\mathrm{ant}}$ are concatenated in the first dimension and ordered according to the minimum Euclidean distance to the desired location $\mathbf{\Gamma}_\mathrm{r}$. The desired location vector $\mathbf{\Gamma}_\mathrm{r} \in \mathbb{R}^{1\times 3}$ and the prior location matrix $\mathbf{\Gamma}_\mathrm{p}^S \in \mathbb{R}^{S\times 3}$ are extended by repeating their coordinates until $\mathbf{\Gamma}_\mathrm{p}^S \in \mathbb{R}^{S\times N_\mathrm{sub}}$ and $\mathbf{\Gamma}_\mathrm{r} \in \mathbb{R}^{1\times N_\mathrm{sub}}$. Hence, the complete location matrix is formed as $\mathbf{H}_\mathrm{LOC} = [\mathbf{\Gamma}_\mathrm{r} \sqcup_1 \mathbf{\Gamma}_\mathrm{p}^S] \in \mathbb{R}^{(S+1)\times N_\mathrm{sub}}$. Finally, the conditional input to the cGAN is constructed as

$$\mathcal{H}_c = \left[\left(\mathbf{H}_N \sqcup_1 \mathcal{H}_\mathrm{p}^S\right) \sqcup_3 \mathbf{H}_\mathrm{LOC}\right] \in \mathbb{R}^{(S+1)\times N_\mathrm{sub}\times(2N_\mathrm{ant}+1)}, \tag{12}$$

where $\mathbf{H}_N \in \mathbb{R}^{N_\mathrm{sub}\times 2N_\mathrm{ant}}$ is a matrix of random values drawn from a Gaussian distribution. The desired channel $\mathbf{H}_r$ is recreated in the $\mathbf{H}_N$ position at the generator output.

Ideally, the true recreated CSI $\mathcal{H}_\mathrm{r} \in \mathbb{R}^{(S+1)\times N_\mathrm{sub}\times(2N_\mathrm{ant}+1)}$ is found in the output of the cGAN. Hence, each d time snapshot in $\mathbf{H}(d) = \mathcal{H}(d,:,:)$ is used as ground truth value for $\mathbf{H}_r$, the CSI recreated at the desired location $\mathbf{\Gamma}_\mathrm{r}$. The pre-processing of the labels for the cGAN include:

- $\mathbf{H}(d) \in \mathbb{C}^{N_\mathrm{sub}\times N_\mathrm{ant}}$ is multiplied by a scaling factor.
- $\mathbf{H}_r = [\mathfrak{Re}\{\mathbf{H}(d)\} \sqcup_2 \mathfrak{Im}\{\mathbf{H}(d)\}] \in \mathbb{R}^{N_\mathrm{sub}\times 2N_\mathrm{ant}}$ is the desired real valued CSI at location $\mathbf{\Gamma}_\mathrm{r}$.

Finally, the label for the semi-supervised learning of the cGAN is constructed as

$$\mathcal{H}_\mathrm{r} = \left[\left(\mathbf{H}_r \sqcup_1 \mathcal{H}_\mathrm{p}^S\right) \sqcup_3 \mathbf{H}_\mathrm{LOC}\right] \in \mathbb{R}^{(S+1)\times N_\mathrm{sub}\times(2N_\mathrm{ant}+1)}, \tag{13}$$

where the prior-CSI $\mathcal{H}_\mathrm{p}^S$, the location matrix $\mathbf{H}_\mathrm{LOC}$ and the recreated CSI $\mathbf{H}_r$ form the desired output. We refer to the generator output as $\tilde{\mathcal{H}}_\mathrm{r} = G(\mathcal{H}_\mathrm{c}) \in \mathbb{R}^{(S+1)\times N_\mathrm{sub}\times(2N_\mathrm{ant}+1)}$, where G is the generator mapping function that tries to approximate the label $\mathcal{H}_\mathrm{r}$. Figure 4 summarizes the process of configuring the conditional input and the desired output for the generator NN. The discriminator D is a classifier for which the inputs and labels are, respectively, $\mathcal{H}_\mathrm{r} \to \{\text{true}\}$ and $\tilde{\mathcal{H}}_\mathrm{r} \to \{\text{fake}\}$.

B. ADVERSARIAL NETWORK ARCHITECTURE

Figure 3 shows the interconnection between the generator and the discriminator NNs for the adversarial training. Here, the generator NN consists of a U-shaped deep NN (U-Net) [31] which has two paths for the flow of information between blocks: the encoder-decoder path and the skip connections path, see Figure 5. The discriminator NN consists of a Patch-NN [31] where the input is reduced to a patch of arbitrary size; then, each coefficient of the patch is classified as real or fake.

Figure 5 shows the U-Net architecture employed for the generator, the $N_g/2$ downsampling blocks for the encoder and the $N_g/2$ upsampling blocks for the decoder, where N_g is the total number of processing units. Each downsampling block consists of one convolutional 2-dimensional layer (Conv2D), one batch normalization layer (BatchNorm), and a leaky rectifier linear unit (LeakyReLU) activation function, where $y = x$ for $x > 0$, and $y = 0.3x$ for $x < 0$. Each

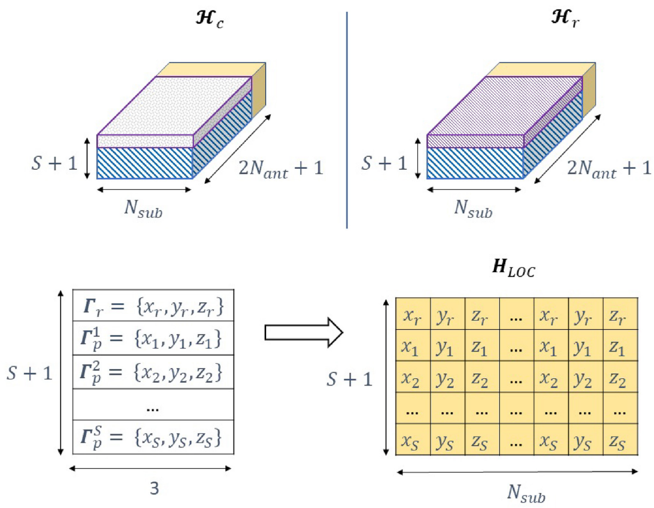

FIGURE 4. Organization of the S prior-CSI estimates $\mathcal{H}_\mathrm{p}^S$, their locations Γ_p^S and the desired channel H_r in the desired location Γ_r at the input $\mathcal{H}_\mathrm{c}$ and the output $\mathcal{H}_\mathrm{r}$ of the cGAN.

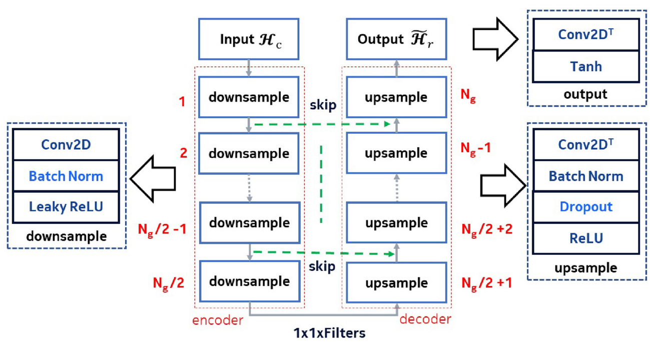

FIGURE 5. U-Net architecture deployed as the generator including encoder and decoder pipeline and numbering for skip connections.

upsampling block consists of one transposed convolutional 2-dimensional layer (Conv2D^T), followed by BatchNorm and ReLU as activation function. A dropout layer [32] can be included in some of the upsampling blocks to avoid overfitting. The skip connection illustrated in Figure 5 happens between the output of the n^{th} downsampling block and the output of the $(N_g - n)^{\text{th}}$ upsampling block, where $n = [1, 2, \ldots, (N_g/2 - 1)]$. Those skip connections provide more information to the decoder block [31] since the input to each upsampling block is the concatenation $\mathcal{X}_{(N_g-n+1)} = [\mathcal{Y}_n \sqcup_3 \mathcal{Y}_{(N_g-n)}]$, where $\mathcal{Y}_n$ is the output of the n^{th} block.

For the discriminator NN, we employ a Patch-NN [31] as depicted in Figure 6. First, downsampling blocks are used to reduce the dimensionality of the input signal to some patch of arbitrary size. Second, the patch is processed by a sequence of convolutional layers (Conv2D + BatchNorm + LeakyReLU and Conv2D + Linear). Then, the discriminator is trained to classify each patch coefficient as real or fake. Implementation details are provided in Section V.

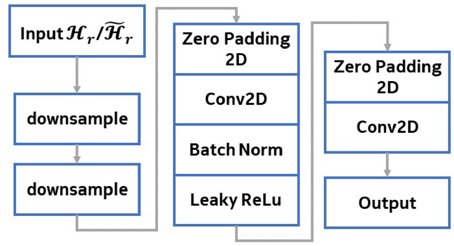

FIGURE 6. Patch-Net architecture deployed for the discriminator.

C. OPTIMIZATION WITH cGAN

As shown in Figure 3, in a cGAN there are two NNs playing a minmax game where the generator $G : \{\mathcal{H}_\mathrm{c}\} \rightarrow \mathcal{H}_\mathrm{r}$ tries to fool the discriminator $D : \{\tilde{\mathcal{H}}_\mathrm{r}\} \rightarrow \{\text{true}\}$, and it is conditional because some prior knowledge is provided. Mathematically, the optimization objective of a cGAN has two terms

$$G^* = \arg\min_G \max_D \mathcal{L}_{\mathrm{cGAN}}(G, D) + \alpha \mathcal{L}_{\mathrm{L}_2}, \tag{14}$$

where $\mathcal{L}_{\mathrm{cGAN}}(G, D)$ is the adversarial loss, $\mathcal{L}_{\mathrm{L}_2}$ is the L2 loss, and α is the weighting factor [31]. The adversarial loss is computed as

$$\begin{aligned}\mathcal{L}_{\mathrm{cGAN}}(G, D) = \mathbb{E}\big[\log D(\mathcal{H}_\mathrm{r})\big] \\ + \mathbb{E}\big[\log(1 - D(G(\mathcal{H}_\mathrm{c}))\big],\end{aligned} \tag{15}$$

where the generator G learns to map the input data $\mathcal{H}_\mathrm{c}$ to the output data $\mathcal{H}_\mathrm{r}$ such that $\tilde{\mathcal{H}}_\mathrm{r} = G^*(\mathcal{H}_\mathrm{c})$, and the discriminator D tries to recognize the channels generated by G. In order to have the generated output wireless channels $\tilde{\mathcal{H}}_\mathrm{r}$ close to the wireless channel labels $\mathcal{H}_\mathrm{r}$, a weighted L_2 loss

$$\mathcal{L}_{\mathrm{L}_2}(G) = \mathbb{E}[\|\mathcal{H}_\mathrm{r} - G(\mathcal{H}_\mathrm{c})\|_\mathrm{F}] \tag{16}$$

is included as a regularization term.

The generator and the discriminator NNs have each their own optimizer with a chosen learning rate that defines how fast the weights of a NN should change according to the computed gradient. Both the generator and the discriminator NNs have their gradients updated in each epoch, which may lead to instabilities during training. Hence, small learning rates are usually considered. For testing, or inference, only the generator architecture is used. Therefore, only knowledge of $\mathcal{H}_\mathrm{c}$ is needed. In practice, at inference time, we are able to estimate/predict a channel based on its location and the prior-knowledge provided by the UNNs.

V. SIMULATIONS AND RESULTS

In order to train and test our CSI recreation framework, we use the MaMIMO dataset provided by KU Leuven [33]. The MaMIMO dataset is a set of indoor MIMO channel measurements collected under line of sight (LOS) and non-line of sight (NLOS) conditions with recordings of the measurement location relative to the BS. Table 1 presents the configuration of the MaMIMO measurement campaign. From the

TABLE 1. Characteristics of the MaMIMO indoor measurement campaign [33].

Carrier frequency	2.61 GHz
Bandwidth	20 MHz
Total OFDM subcarriers	100
BS antenna	URA 8×8
UE antenna	dipole
Normalized received signal strength	$[-10, -1]$ dB
Location label accuracy	1 mm [34]

TABLE 2. Description of the UNN structures.

Name	Input tensor	L	$k_{1:L-1}$	k_L	Upsampling matrices	Output size
UNN 64	$\mathcal{Z}_0 \in \mathbb{R}^{4\times4\times64}$	6	64	128	$\mathbf{A}_{\{1:L-2\}}, \mathbf{C}_{\{1:L-2\}}$	$\mathcal{H}_{\mathrm{p}} \in \mathbb{R}^{64\times64\times128}$
UNN 16	$\mathcal{Z}_0 \in \mathbb{R}^{4\times4\times16}$	6	16	128	$\mathbf{A}_{\{1:L-4\}}, \mathbf{C}_{\{1:L-4\}}$	$\mathcal{H}_{\mathrm{p}} \in \mathbb{R}^{16\times64\times128}$

MaMIMO dataset, we select channels taken in an area of 1255 mm by 85 mm, such that $\mathcal{H} \in \mathbb{R}^{4267\times64\times128}$ there are a total of 4267 channel measurements spaced 5mm from each other. From the entire $\mathcal{H}$ dataset, a few measurements are used as $\mathcal{H}_{\mathrm{mes}}$ for adapting the UNN CSI estimator. The remaining part of the dataset $\mathcal{H}$ is used as the ground truth $\mathbf{H}_{\mathrm{r}}$ for training and testing the cGAN CSI recreation. We use the normalized squared error (NSE) NSE $= \frac{\|\mathbf{B}-\tilde{\mathbf{B}}\|_F^2}{\|\mathbf{B}\|_F^2}$ as our performance metric for CSI recreation.

In order to show the power and the pitfalls of UNNs and motivate the choice of combining UNNs with cGANs, we present our results following a series of experiments. First, we define a UNN CSI estimator to recover $\mathcal{H}_{\mathrm{p}}$ with an SNR $\approx$ 20 dB. Second, we show the limited interpolation capabilities of UNNs. Third, we present the CSI recreation with a cGAN for different settings of prior-CSI. Finally, we evaluate the dependencies of the models on the location matrix and the impact of inaccuracies on estimating the desired location $\mathbf{\Gamma}_{\mathrm{r}}$ on the performance of the cGAN.

In Table 2 we define the UNN structures that we use to derive the optimal weights for each small set of measurements $\mathcal{H}_{\mathrm{mes}}$, where UNN-64 estimates 64 time snapshots and UNN-16 estimates 16 time snapshots jointly. For CSI estimation with UNNs, we choose $\mathcal{H}_{\mathrm{mes}}$ to have time snapshots every 15 mm, such that a UNN-64 estimates the CSI for a sequence of measurements covering 64×15 mm $=$ 960 mm ($\approx 8.35\lambda$), and a UNN-16 covers 240 mm ($\approx 2.08\lambda$). The input tensor $\mathcal{Z}_0$ is drawn from a uniform distribution as $U(-0.15, +0.15)$ and kept fixed for UNN CSI-estimation. After setting the UNN structure, the trainable parameters $\mathcal{K}$ are initialized from random values and $I = 25000$ gradient updates are performed, using an Adam optimizer [32] with a learning rate of 0.01, to find the best $\mathcal{K}^*$ for each set of measurements, separately. Figure 7 presents the cumulative distribution function (CDF) of the NSE for CSI estimation for the two UNN structures in Table 2 at LOS and NLOS propagation conditions. The number of prior sets indicates how many different measurement areas are considered. Note that, if the channel measurements have the same dimensionality, a UNN structure P can be reused, but a different set of UNN parameters $\mathcal{K}^*$ needs to be

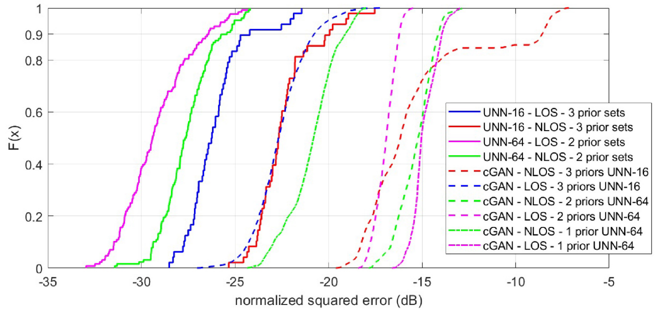

FIGURE 7. Cumulative distribution function $F(x)$ of the normalized squared error for UNN and cGAN results for CSI recreation.

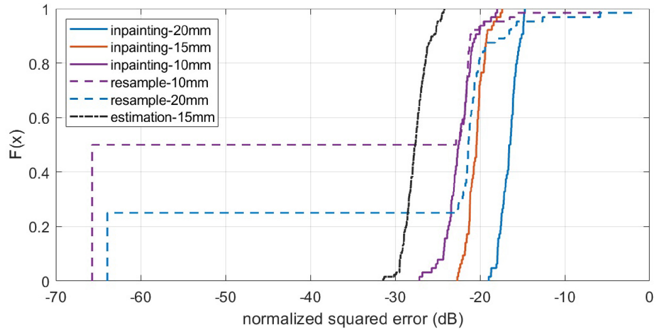

FIGURE 8. Cumulative distribution function $F(x)$ of the normalized squared error for NLOS channel interpolation using UNN-64 in inpainting mode. Best case and worst case using resampling function from MATLAB are plotted for reference. The performance of UNN-64 for channel estimation is also plotted for reference.

found for each different area where $\mathcal{H}_{\mathrm{mes}}$ is collected [17]. From Figure 7 we can observe that the UNN-64 has a better performance than UNN-16, this is expected since UNN-64 has more parameters and also access to more channel snapshots than a UNN-16. Moreover, for the same UNN structure, LOS channel measurements have better estimation performance than NLOS channel measurements. The LOS channels have smoother variations; therefore, the UNN takes better advantage of the correlations between the measured snapshots which leads to better estimation results. Overall, our UNN CSI-estimators provide more than 10 dB estimation gain if compared with the normalized received signal strength (RSS) of the MaMIMO measurements [33], see Table 1.

For UNN based CSI interpolation, we use the UNN-64 presented in Table 2 and follow the loss function in equation (10). Since the MaMIMO dataset has a resolution of 5 mm, the UNN CSI-interpolator is iterated to recover the channel measurements at 5 mm spacing from each other. Hence, for this experiment, the UNN-64 covers an area of 64×5 mm $= 320$ mm ($\approx 2.78\lambda$). The mask $\mathcal{M}$ is used to reduce the resolution of $\mathcal{H}_{\mathrm{mes}}$ such that the resolution of the starting channel measurements $\mathcal{H}'_{\mathrm{mes}}$ vary between 10 mm to 20 mm. This is the distance between each time snapshot. In the first row of Figure 9, we present $\mathcal{H}_{\mathrm{mes}}$ on the left and $\mathcal{H}'_{\mathrm{mes}}$ on the right for a single antenna element with initial measurement resolution of 10 mm. Figure 10 has the same

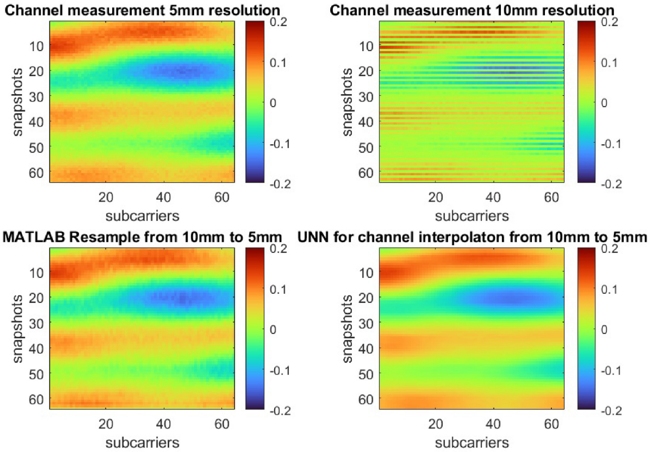

FIGURE 9. Comparison of the results for interpolation by a factor of 2 with the UNN CSI interpolator and resampling from MATLAB. On the first row left, there is a plot of the expected 5 mm resolution measurement $\mathcal{H}_{\text{mes}}$ for one antenna element. On the first row right, there is a plot of the initial measurement $\mathcal{H}'_{\text{mes}}$ with 10 mm resolution. The second row present the results of interpolation with resampling function from MATLAB on the left and the UNN CSI interpolator on the right. On the *x*-axis there is the number of subcarriers, and on the *y*-axis there is the number of time snapshots.

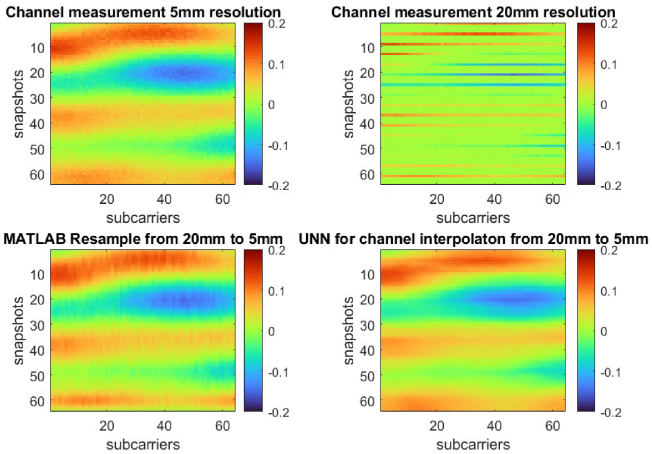

FIGURE 10. Comparison of the results for interpolation by a factor of 4 with the UNN CSI interpolator and resampling from MATLAB. On the first row left, there is a plot of the expected 5 mm resolution measurement $\mathcal{H}_{\text{mes}}$ for one antenna element. On the first row right, there is a plot of the initial measurement $\mathcal{H}'_{\text{mes}}$ with 20 mm resolution. The second row present the results of interpolation with resampling function from MATLAB on the left and the UNN CSI interpolator on the right. On the *x*-axis there is the number of subcarriers, and on the *y*-axis there is the number of time snapshots.

structure, but for $\mathcal{H}'_{\text{mes}}$ with an initial resolution of 20 mm which requires an upsampling by a factor of 4 to derive channels every 5 mm. The input tensor $\mathcal{Z}_0$ is initially drawn from a uniform distribution $U(-0.15, +0.15)$. After every $I_t = 5000$ gradient iterations, the input tensor $\mathcal{Z}_0$ is summed with a random Gaussian noise with zero mean and a standard deviation of 0.4. The channel interpolation is finished after $I = 25000$ gradient updates are performed, using the Adam optimizer [32] with 0.01 learning rate. Figure 8 shows the results of the UNN CSI interpolator for the NLOS dataset iterated using the inpainting procedure in Section III-D. We can observe that the performance of the UNN CSI interpolator degrades quickly when we increase the spacing between the known measurement snapshots $\mathcal{H}'_{\text{mes}}$. For comparison, we plot the best and worst case results using the MATLAB built-in function 'resample', with a FIR anti-aliasing filter of order $20p$ with $p = [2, 3, 4]$. Figures 9 and 10 present in their second row the results of the interpolation with the resample function from MATLAB on the left, and the UNN CSI interpolator on the right after upsampling by a factor of 2 (10 mm) and 4 (20 mm), respectively. From Figure 8, it can be observed that the UNN based CSI interpolator has a comparable performance as using the resample function from MATLAB only for the case of increasing the resolution by a factor of 2, from 10 mm to 5 mm in the purple curve. The perfect results in the purple and blue dashed lines correspond to the reconstruction error of the coefficients which are used for interpolation. The UNN based CSI interpolator defines a sequence of operations that impacts the known coefficients. Hence, there are no perfect results in the CDF curve of the UNN based CSI interpolator. The black dashed line denotes the performance of the UNN CSI estimator when the channels are known every 15 mm. For the same UNN structure, there is a performance degradation of about a 6 dB if we use it for CSI-interpolation instead of CSI-estimation. Due to its performance degradation and the need to re-iterate the parameters, the UNN CSI interpolator is not our choice for CSI recreation. Nonetheless, we use the UNN CSI estimator to generate prior-CSI due to its great performance, but only to a small set of measurements to avoid latency overhead with the gradient iterations. Note that in this experiment we have considered measurements taken along a line and performed interpolation along the same direction. For CSI recreation, however, we do not assume a regular grid of measured priors. Hence, the resample function is not applicable.

Two sets of UNN-64 parameters are computed using equation 8 for CSI estimation which represents a pool of 128 prior-CSI estimates. For UNN-16, three sets of parameters are computed using equation 8 which represent a pool of 48 prior-CSI estimates. We assume the prior-CSI locations $\boldsymbol{\Gamma}_{\text{p}}$ are known. In our case, it is provided by the dataset owner. In a practical implementation, we could derive the location from the prior-CSI estimates by Unitary Tensor ESPRIT [28], for instance. Then, for CSI recreation with a cGAN, a sub-set of $S = 3$ CSI-location pairs are selected according to their minimum Euclidean distance to the location $\boldsymbol{\Gamma}_{\text{r}}$ where we aim to recreate the CSI. Hence, the input to our cGAN is $\mathcal{H}_c \in \mathbb{R}^{4\times 64\times 129}$, where 64 is the number of subcarriers. The architecture details of our generator and discriminator are presented in Table 3 and Table 4, respectively. The adversarial training runs for 150 epochs with 60% of the simulated channels used for training and 40% for testing. Adam optimizers with a learning rate of 2×10^{-4} are used for the discriminator and the generator NNs.

Figure 7 presents the CDF curves of the NSE for CSI recreation using a cGAN which has prior-knowledge and the desired location as its input. The results for LOS and NLOS channel measurements are provided. Overall, the

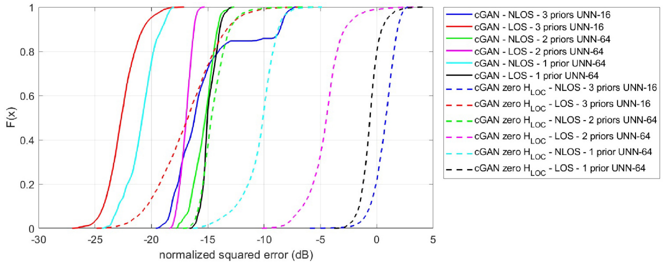

FIGURE 11. Comparison of the importance of the location matrix $\mathbf{H}_{\text{LOC}}$ for each trained cGAN model. The full lines are the CDFs $F(x)$ when there is full knowledge of $\mathbf{H}_{\text{LOC}}$. After training, $\mathbf{H}_{\text{LOC}} = 0$ and the priors are selected at random. These CDF results are shown in dashed lines.

TABLE 3. Description of the U-Net deployed as generator NN.

	Block	N_{filter}	Filter size	Stride	Padding	BatchNorm	Dropout	Activation
1	downsample	64	[3,3]	[1,1]	Yes	No	No	LeakyReLU
2	downsample	64	[1,4]	[1,2]	Yes	Yes	No	LeakyReLU
3	downsample	64	[1,4]	[1,2]	Yes	Yes	No	LeakyReLU
4	downsample	64	[1,4]	[1,2]	Yes	Yes	No	LeakyReLU
5	downsample	64	[1,4]	[1,2]	Yes	Yes	No	LeakyReLU
6	downsample	128	[4,4]	[1,1]	No	Yes	No	LeakyReLU
7	upsample	64	[4,4]	[2,2]	No	Yes	Yes	ReLU
8	upsample	64	[1,4]	[1,2]	Yes	Yes	Yes	ReLU
9	upsample	64	[1,4]	[1,2]	Yes	Yes	Yes	ReLU
10	upsample	64	[1,4]	[1,2]	Yes	Yes	No	ReLU
11	upsample	64	[1,4]	[1,2]	Yes	Yes	No	ReLU
12	upsample	64	[3,3]	[1,1]	Yes	Yes	No	ReLU
13	output	129	[1,1]	[1,1]	Yes	No	No	TanH

TABLE 4. Description of the Patch-Net deployed as discriminator NN.

Block	N_{filter}	Filter size	Stride	Padding	BatchNorm	Activation
downsample	128	[3,3]	[1,1]	Yes	Yes	LeakyReLU
downsample	128	[2,4]	[1,2]	Yes	Yes	LeakyReLU
zero padding 2D	-	-	-	Yes	-	-
Conv2D	256	[3,3]	[1,1]	No	Yes	LeakyReLU
zero padding 2D	-	-	-	Yes	-	-
Conv2D	1	[3,3]	[1,1]	No	No	Linear

cGAN has a performance degradation if compared with UNNs. However, the cGAN is recreating channels on a non-regular grid where the distance between the prior-CSI and the desired channel is variable. It can be observed that CSI recreation for LOS propagation has a better performance, see blue dashed line. Furthermore, how the prior-CSI estimates are selected also influences the performance of the CSI recreation. Figures 12-17 present the NSE of the recreated CSI according to their position in the propagation environment. The places where the prior-CSI estimates were selected for the UNNs are highlighted in dark blue. Our best cGAN model for LOS, the blue dashed line in Figure 7, has a 90% performance of about -20.76 dB and uses only 48 prior channels which are sampled in a diagonal manner, see Figure 12. Nonetheless, for NLOS this set of priors provides the worst model performance, the red dashed line in Figure 7, with a 90% performance of about -8.58 dB. In Figure 15, we can observe that the cGAN has difficulty in recreating channels in the transition area between the first and second set of priors. For the NLOS measurements, the best approach to collect priors is to use a single sequence of measurements as in Figure 17 which provides a 90% performance of about -19.1 dB, green dash-point line in Figure 7. The best prior-CSI sampling for the LOS and the NLOS cases are quite different, for the LOS they should be spread within a small number of time snapshots (see Figure 12), while for the NLOS they should be sequential (see Figure 17). We reason that a LOS environment has smooth transitions (small variability), so that the cGAN takes better advantage in knowing the channel at rather spaced locations. On the other hand, NLOS environments have more variations and the cGAN performs better when it observes how the channel evolves for longer periods. However, it seems that there is a limit on how many priors should be provided as the results for the LOS and the NLOS environments based on two sets of priors degrade the performance if compared with the best case in each environment.

We also evaluate the importance of the location matrix $\mathbf{H}_{\text{LOC}}$ for the cGAN to recreate the CSI. Given the trained model, the experiment consists in setting the location matrix to zero $\mathbf{H}_{\text{LOC}} = \mathbf{0}$. Since the location is not available, the prior-CSI to the cGAN input are selected at random. Figure 11 presents the results of this experiment and provides a comparison with the CSI recreation with perfect knowledge of location. It can be observed that nearly all cGAN models have a large performance degradation when $\mathbf{H}_{\text{LOC}} = \mathbf{0}$. This indicates that the model was able to learn some relationship between the channels and their positions. Nonetheless, the performance degradation for the cGAN trained for NLOS channels with two sets of priors is of about 1 dB. In this case, the location matrix has a low importance and the cGAN can output a reasonable CSI recreation knowing any set of 3 priors. This result also contributes to our understanding that providing more priors does not translate to a better performance. In order to observe how inaccuracies on estimating $\mathbf{\Gamma}_{\text{r}}$ impact the performance of our CSI recreation method, we include Gaussian noise to the desired location $\tilde{\mathbf{\Gamma}}_{\text{r}}$ and do not re-train the cGAN model. The location of the prior-CSI $\mathbf{\Gamma}_{\text{p}}^{S}$ are assumed to be accurate as the UE collects many time snapshots for prior-CSI estimation, i.e., at least 16 time snapshots for our experiments. The inaccuracy in estimating the desired location is modeled by a Gaussian distribution with mean $\mu = \{0, 1, 2, 10, 15, 20\}$ and standard deviation $\sigma = 0.5$. Figure 18 presents the histogram of the Euclidean distance between the correct desired location $\mathbf{\Gamma}_{\text{r}}$ and the estimated desired location $\tilde{\mathbf{\Gamma}}_{\text{r}}$ for each inaccuracy distribution. Figure 19 presents the CDF of the NSE for the NLOS case with a single set of prior-CSI for the different location inaccuracies $\tilde{\mathbf{\Gamma}}_{\text{r}}$. It can be observed that when the location inaccuracy is around 1 cm (cyan), the performance of the CSI recreation does not change. This is due to the fact that the prior-CSI are taken 15 mm from each other. Hence, a location estimation error below 1.5 cm does not change the combination of prior-CSI used by the cGAN. Increasing the desired location error $\tilde{\mathbf{\Gamma}}_{\text{r}}$ also increases the 90% accuracy for recreating the CSI (red, purple and black). However, it has a limit as desired location estimates $\tilde{\mathbf{\Gamma}}_{\text{r}}$ with error above 26 cm causes degradation of the CSI recreation performance. Figure 20 presents the CDF of the NSE for the LOS case with three sets of prior-CSI for different location inaccuracies $\tilde{\mathbf{\Gamma}}_{\text{r}}$. As in the NLOS case, the performance

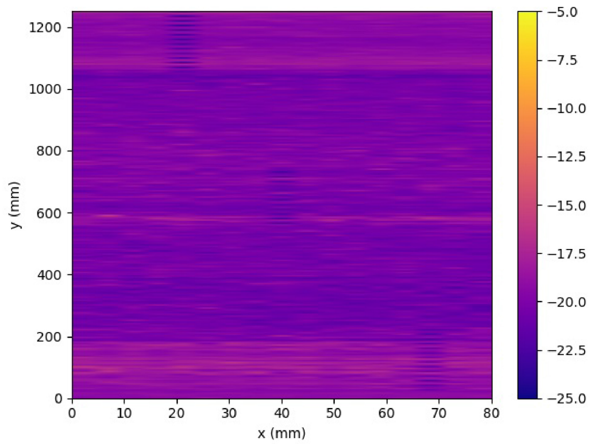

FIGURE 12. **CSI recreation performance map for LOS measurements with 3 sets of prior-CSI estimated by a UNN 16 structure. The cGAN is trained with the $\mathcal{L}_{L_2}$ loss. The colors represent the NSE in dB values.**

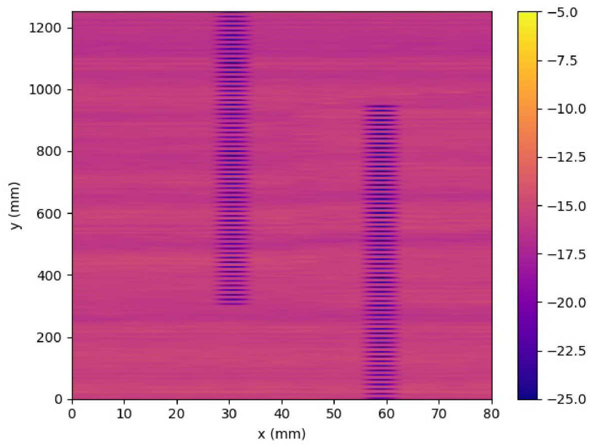

FIGURE 13. **CSI recreation performance map for LOS measurements with 2 sets of prior-CSI estimated by a UNN 64 structure. The cGAN is trained with the $\mathcal{L}_{L_2}$ loss. The colors represent the NSE in dB values.**

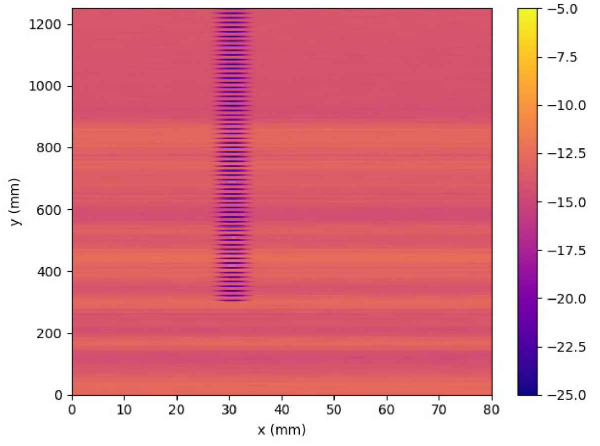

FIGURE 14. **CSI recreation performance map for LOS measurements with 1 set of prior-CSI estimated by a UNN 64 structure. The cGAN is trained with the $\mathcal{L}_{L_2}$ loss. The colors represent the NSE in dB values.**

of CSI recreation does not change when the error in the desired location is below 1.5 cm. Some performance gain can be achieved at the 70% accuracy level when increasing

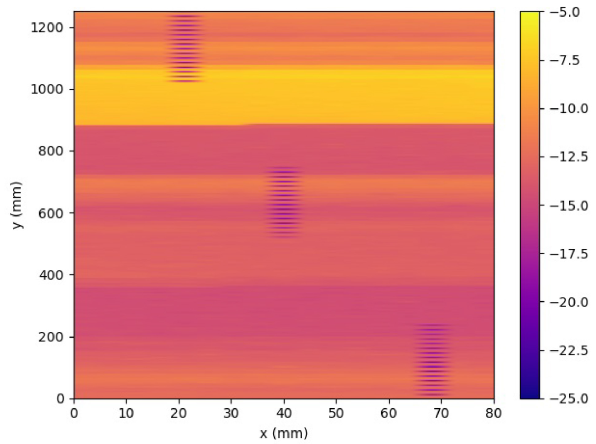

FIGURE 15. **CSI recreation performance map for NLOS measurements with 3 sets of prior-CSI estimated by a UNN 16 structure. The cGAN is trained with the $\mathcal{L}_{L_2}$ loss. The colors represent the NSE in dB values.**

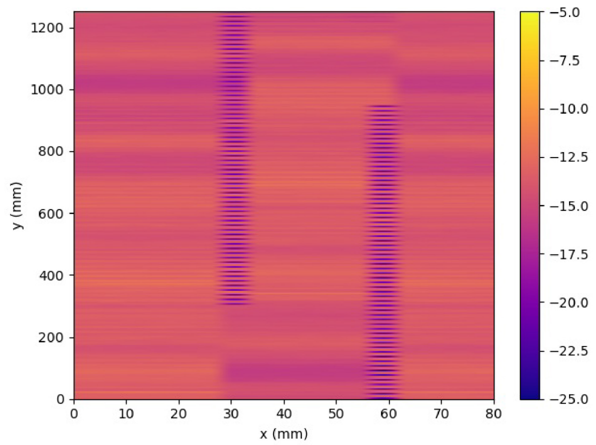

FIGURE 16. **CSI recreation performance map for NLOS measurements with 2 sets of prior-CSI estimated by a UNN 64 structure. The cGAN is trained with the $\mathcal{L}_{L_2}$ loss. The colors represent the NSE in dB values.**

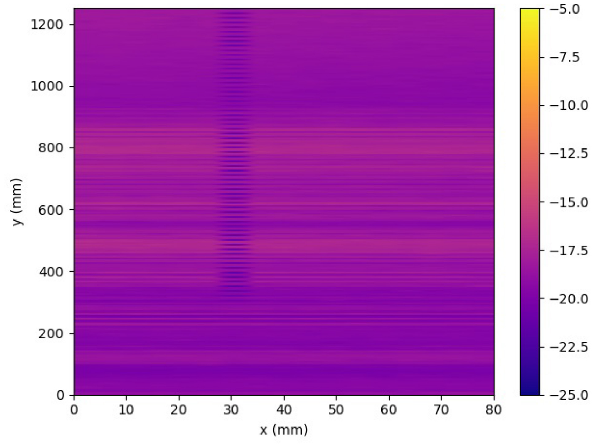

FIGURE 17. **CSI recreation performance map for NLOS measurements with 1 set of prior-CSI estimated by a UNN 64 structure. The cGAN is trained with the $\mathcal{L}_{L_2}$ loss. The colors represent the NSE in dB values.**

the desired location error $\tilde{\mathbf{\Gamma}}_{\mathrm{r}}$. However, this gain is not as large as in the NLOS case because the prior-CSI are already distributed in the best LOS scenario. Increasing the location

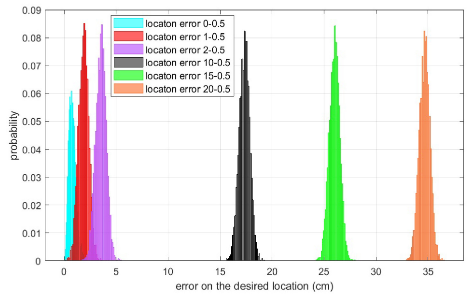

FIGURE 18. Histogram of the Euclidean distance between Γ_r and $\tilde{\Gamma}_r$. The estimation error is drawn from a Gaussian distribution with μ-σ described in the legend of the histogram.

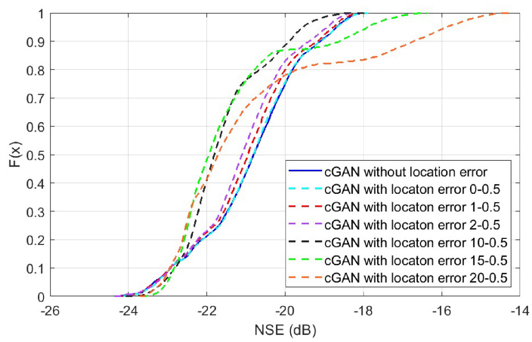

FIGURE 19. Cumulative distribution function $F(x)$ of the NSE for CSI recreation with error in the desired location $\tilde{\Gamma}_r$ for the best NLOS case (Figure 17) without retraining.

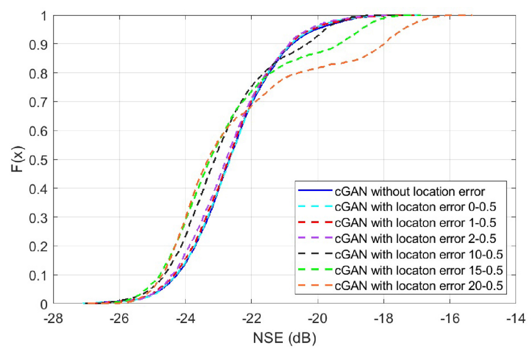

FIGURE 20. Cumulative distribution function $F(x)$ of the NSE for CSI recreation with error in the desired location $\tilde{\Gamma}_r$ for the best LOS case (Figure 12) without retraining.

estimation error and improving the performance might sound counter-intuitive. However, we should point out that this experiment is performed at inference time, without retraining the cGAN. Hence, error in estimating the desired location can change the set of prior-CSI that the cGAN is taking at its input, but does not change the statistical distribution modeled by the cGAN. Those results indicate that selecting the prior-CSI according to their minimum Euclidean distance towards the desired location $\boldsymbol{\Gamma}_r$ may not be the best approach for CSI recreation.

Regarding the state of the art, our approach provides a much better performance if compared to the 6 dB reported in [11]. This is mainly due to two reasons, in [11] a much larger area is considered and the authors just rely on supervised learning. It has been reported that the generative loss function improves the richness of the images being generated as L_2 norm tends to generate blurred outputs [30], [31]. Moreover, our cGAN architecture is less complex since we just use 13 layers for the U-Net at the generator while [11] has reported 28 layers to process the images of the environment map and output the wireless channel. The generator described in Table 3 has a total of 552513 trainable parameters which correspond to about 3.16% of the complex channel coefficients $\boldsymbol{\mathcal{H}} \in \mathbb{C}^{4267\times 64\times 64}$ collected in the studied area. Hence, the memory requirement for the digital twin running at the BS side is also reduced. Our cGAN takes about 6 hours to train in a computer with 16 GB of RAM and a GPU with 2 GB of dedicated memory.

Regarding complexity, the UNN-64 structure contains 29312 trainable parameters which correspond to 11.18% of the coefficients in a channel measurement $\boldsymbol{\mathcal{H}}_{\text{mes}} \in \mathbb{C}^{64\times 64\times 64}$. The UNN-16 structure contains 3488 trainable parameters which correspond to 5.32% of the coefficients in a channel measurement $\boldsymbol{\mathcal{H}}_{\text{mes}} \in \mathbb{C}^{16\times 64\times 64}$. Therefore, UNNs are an under-parameterized representation of the wireless channels that provide channel estimation gain in most cases. Due to the underparameterization, we assume that the UEs can compute the UNN parameters based on small measurement campaigns and send them to the BS. The BS trains the cGAN based on the collected prior-CSI. After the cGAN model is trained, the UEs within the representation area can just report their location $\boldsymbol{\Gamma}_r$ and the BS is able to recreate their full explicit CSI. For a person walking in this indoor environment (Table 1) with a velocity of 1 m/s, a CSI report is needed at least every 57 mm in order to meet the Nyquist sampling criteria. Assuming we can use 10 bits to quantize 1 m space with 1 mm accuracy, reporting the coordinates of the desired location $\boldsymbol{\Gamma}_r$ can be very efficient in reducing the CSI feedback overhead. The access to the full characteristics of the wireless channel with low overhead is the main contribution of this paper since we are not able to recover all the channel coefficients in current 3GPP type II CSI reporting. In order to track the reliability of the CSI recreation method, from time to time, i.e., once per day, the BS may send the cGAN model to the UEs. Hence, the UEs can observe the recreated CSI and flag when it is completely wrong. Which, then, may trigger collection of new prior-CSI and re-training of the cGAN. For expanding the operational area of the CSI recreation, we expect that more prior measurements need to be collected. This would increase the number of UNNs, but the cGAN could be re-trained for the enlarged operational area. Nonetheless, we leave this study to future work.

VI. CONCLUSION

In this paper we propose to combine UNNs with cGANs to recreate wireless channels within the digital twin of a propagation environment. The channel is recreated based on prior-CSI estimates from UNNs and the location where we aim to recreate the channels. The CSI recreation allows to access the full characteristics of the wireless channel with low communication overhead, which is not possible with current 3GPP type II CSI reporting. The cGAN is able to recreate the CSI for indoor measurements under LOS and NLOS propagation conditions. We show in our results that the way we select the prior-CSI measurements impacts the performance and that availability of more prior-CSI estimates does not necessarily translates into improved CSI recreation performance. Moreover, we provide results for CSI interpolation with UNNs which has motivated our choice of cGANs as the second AI/ML instance for CSI recreation. In addition, we set up experiments to access how good the cGAN uses the location information to recreate CSI, and how inaccuracies in estimating the desired location impacts the performance of the CSI recreation. Despite their differences, CSI recreation outperforms the state of the art FadeNet and is less complex. Future work may consider outdoor scenarios, increase the CSI recreation area, and study the best approach to select areas where the prior-CSI measurements should be collected.

REFERENCES

[1] H. Viswanathan and P. E. Mogensen, “Communications in the 6G era,” *IEEE Access*, vol. 8, pp. 57063–57074, 2020.

[2] “Advanced plans for 5G.” 3GPP. Jul. 2021. [Online]. Available: https://www.3gpp.org/news-events/2210-advanced_5g

[3] B. Brik, K. Boutiba, and A. Ksentini, “Deep learning for B5G open radio access network: Evolution, survey, case studies, and challenges,” *IEEE Open J. Commun. Soc.*, vol. 3, pp. 228–250, 2022.

[4] “Study on artificial intelligence (AI)/machine learning (ML) for NR air interface,” 3GPP, Sophia Antipolis, France, Rep. TR38.843, 2022.

[5] “What is a digital twin?” IBM. 2022. [Online]. Available: https://www.ibm.com/topics/what-is-a-digital-twin

[6] E. Glaessgen and D. Stargel, “The digital twin paradigm for future NASA and U.S air force vehicles,” in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. 20th AIAA/ASME/AHS Adapt. Struct. Conf. 14th AIAA*, 2012, p. 1818.

[7] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, “5G MIMO data for machine learning: Application to beam-selection using deep learning,” in *Proc. Inf. Theory Appl. Workshop (ITA)*, 2018, pp. 1–9.

[8] M. Dias, A. Klautau, N. González-Prelcic, and R. W. Heath, “Position and LIDAR-aided mmWave beam selection using deep learning,” in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2019, pp. 1–5.

[9] A. Klautau, N. González-Prelcic, and R. W. Heath, “LIDAR data for deep learning-based mmWave beam-selection,” *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 909–912, Jun. 2019.

[10] J. P. T. Borges *et al.*, “Reinforcement learning for scheduling and MIMO beam selection using Caviar simulations,” in *Proc. ITU Kaleidoscope Connecting Phys. Virtual Worlds (ITU K)*, 2021, pp. 1–7.

[11] V. V. Ratnam *et al.*, “FadeNet: Deep learning-based mm-Wave large-scale channel fading prediction and its applications,” *IEEE Access*, vol. 9, pp. 3278–3290, 2021.

[12] V. Savic and E. G. Larsson, “Fingerprinting-based positioning in distributed massive MIMO systems,” in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, 2015, pp. 1–5.

[13] X. Wang, L. Gao, S. Mao, and S. Pandey, “DeepFi: Deep learning for indoor fingerprinting using channel state information,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2015, pp. 1666–1671.

[14] V. Lempitsky, A. Vedaldi, and D. Ulyanov, “Deep image prior,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9446–9454.

[15] R. Heckel and P. Hand, “Deep decoder: Concise image representations from untrained non-convolutional networks,” in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17. [Online]. Available: https://openreview.net/forum?id=rylV-2C9KQ

[16] E. Balevi, A. Doshi, and J. G. Andrews, “Massive MIMO channel estimation with an untrained deep neural network,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2079–2090, Mar. 2020.

[17] B. Vilas Boas, W. Zirwas, and M. Haardt, “Transfer learning capabilities of untrained neural networks for MIMO CSI recreation,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2022, pp. 1288–1293.

[18] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[19] Y. Yang, Y. Li, W. Zhang, F. Qin, P. Zhu, and C.-X. Wang, “Generative-adversarial-network-based wireless channel modeling: Challenges and opportunities,” *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 22–27, Mar. 2019.

[20] T. J. O’Shea, T. Roy, and N. West, “Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks,” in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, 2019, pp. 681–686.

[21] E. Balevi, A. Doshi, A. Jalal, A. Dimakis, and J. G. Andrews, “High dimensional channel estimation using deep generative networks,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 18–30, Jan. 2021.

[22] E. Balevi and J. G. Andrews, “Wideband channel estimation with a generative adversarial network,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 5, pp. 3049–3060, May 2021.

[23] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014, *arXiv:1411.1784*.

[24] X. Li, A. Alkhateeb, and C. Tepedelenlioğlu, “Generative adversarial estimation of channel covariance in vehicular millimeter wave systems,” in *Proc. 52nd Asilomar Conf. Signals, Syst. Comput.*, 2018, pp. 1572–1576.

[25] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, “Channel agnostic end-to-end learning based communication systems with conditional GAN,” in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–5.

[26] A. Smith and J. Downey, “A communication channel density estimating generative adversarial network,” in *Proc. IEEE Cogn. Commun. Aerospace Appl. Workshop (CCAAW)*, 2019, pp. 1–7.

[27] B. Vilas Boas, W. Zirwas, and M. Haardt, “Two-step machine learning approach for channel estimation with mixed resolution RF chains,” in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2021, pp. 1–6.

[28] M. Haardt, F. Roemer, and G. Del Galdo, “Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems,” *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3198–3213, Jul. 2008.

[29] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[30] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.

[31] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[33] A. P. Guevara, S. De Bast, and S. Pollin, “Weave and conquer: A measurement-based analysis of dense antenna deployments,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2021, pp. 1–6.

[34] S. De Bast, A. P. Guevara, and S. Pollin, “CSI-based positioning in massive MIMO systems using convolutional neural networks,” in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, 2020, pp. 1–5.

BRENDA VILAS BOAS (Graduate Student Member, IEEE) received the bachelor's and master's degrees in electrical engineering from the Federal University of Pará (UFPA), Brazil, in 2017 and 2019, respectively. She is currently pursuing the Ph.D. degree in electrical engineering in the area of wireless communications with Nokia and Ilmenau University of Technology, Germany.

She was a Manager of the Community Cellular Telephony (CELCOM) Project with the Telecommunications, Automation and Electronics Research and Development Center (LASSE/UFPA) from 2016 to 2018, which aimed to bridge the digital divide in the Amazon Rain-Forest. She has participated in mobility academic programs in Canada, from 2013 to 2014 and Spain, from 2016 to 2017. Her research interests include machine learning and its application to the wireless physical layer, digital signal processing, and the impact of technology in society.

Ms. Vilas Boas has been a Volunteer of IEEE since 2015. She was a Co-Founder in 2018 and an Advisor from 2018 to 2019 of the IEEE WIE UFPA Student Affinity Group, a Co-Founder in 2015 and the Chair of the IEEE ComSoc UFPA Student Chapter in 2018, the Student Section Representative in 2017 and the Student Activity Chair from 2018 to 2019 for the IEEE Centro-Norte Brazil section. In 2019, she was awarded the IEEE R9 Student and Young Professional Activities Prominent Supporter Award due to her engagement and support to student's and YP's activities in Latin America.

WOLFGANG ZIRWAS (Senior Member, IEEE) received the diploma degree in communication technologies from the Technical University of Munich in 1987. He is currently pursuing the Ph.D. degree with the Ilmenau University of Technology.

In 1987, he started his work with the Siemens Munich Central Research Lab for Communication and RF Technologies and is currently with Nokia Bell Labs, Munich, Germany. Since 1996, he has been involved in BMBF and EU funded Projects like ATMmobil, COVERAGE, 3GET, Scalenet, WINNER, ARTIST4G, METIS, and Fanatasic5G working on topics like optical fibre systems, OFDM, multihop, MIMO, massive MIMO, cooperative antennas, and channel estimation plus prediction. He is currently working on AI/ML based methods for the PHY layer air interface in the context of the German funded Project FunKI. He is the coauthor of four books (beside many journals and conference papers).

Mr. Zirwas received the Best Paper Award in 2012 from the Future Network and Mobile Summit. He was the TCP of several conferences like European Wireless, PIMRC or WDN-CN2012 and an Editor of the *Scientific World Journal.*

MARTIN HAARDT (Fellow, IEEE) studied electrical engineering from Ruhr-University Bochum, Germany, and Purdue University, USA, and received the Diplom-Ingenieur (M.S.) degree from Ruhr-University Bochum in 1991, and the Doktor-Ingenieur (Ph.D.) degree from the Munich University of Technology in 1996.

In 1997, he joined Siemens Mobile Networks, Munich, Germany, where he was responsible for strategic research for third generation mobile radio systems. From 1998 to 2001 he was the Director for International Projects and University Cooperations in the mobile infrastructure business of Siemens, Munich, where his work focused on mobile communications beyond the third generation. Since 2001, he has been a Full Professor with the Department of Electrical Engineering and Information Technology and the Head of the Communications Research Laboratory, Ilmenau University of Technology, Germany. His research interests include wireless communications, array signal processing, high-resolution parameter estimation, and tensor-based signal processing.

Prof. Haardt has received the 2009 Best Paper Award from the IEEE Signal Processing Society, the Vodafone Innovations Award for outstanding research in mobile communications, the ITG Best Paper Award from the Association of Electrical Engineering, Electronics, and Information Technology, and the Rohde and Schwarz Outstanding Dissertation Award. He has been served as a Senior Editor for the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING since 2019, as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2002 to 2006 and from 2011 to 2015, the IEEE SIGNAL PROCESSING LETTERS from 2006 to 2010, the *Research Letters in Signal Processing* from 2007 to 2009, the *Journal of Electrical and Computer Engineering* (Hindawi) since 2009, the *EURASIP Signal Processing* from 2011 to 2014, and as a Guest Editor for the *EURASIP Wireless Communications and Networking*. From 2011 to 2019 he was an Elected Member of the Sensor Array and Multichannel Technical Committee of the IEEE Signal Processing Society, where he served as the Vice Chair from 2015 to 2016, the Chair from 2017 to 2018, and the Past Chair in 2019. Since 2020, he has been an Elected Member of the Signal Processing Theory and Methods Technical Committee of the IEEE Signal Processing Society. Moreover, he has served as the Technical Co-Chair of PIMRC 2005 in Berlin, Germany, ISWCS 2010 in York, U.K., the European Wireless 2014 in Barcelona, Spain, as well as the Asilomar Conference on Signals, Systems, and Computers 2018, USA, and the General Co-Chair of WSA 2013 in Stuttgart, Germany, ISWCS 2013 in Ilmenau, Germany, CAMSAP 2013 in Saint Martin, French Antilles, WSA 2015 in Ilmenau, SAM 2016 in Rio de Janeiro, Brazil, CAMSAP 2017 in Curacao, Dutch Antilles, SAM 2020 in Hangzhou, China, and as well as the Asilomar Conference on Signals, Systems, and Computers 2021, USA.