Fast Channel Estimation by Infinite Width Convolutional Networks

Guillaume Villemaud, Mohammed Mallik

Electronics Letters

LETTER OPEN ACCESS

Fast Channel Estimation by Infinite Width Convolutional Networks

Guillaume Villemaud | Mohammed Mallik

INSA Lyon, Inria, CITI, Villeurbanne, France

Correspondence: Mohammed Mallik (mohammed.mallik@insa-lyon.fr)

Received: 29 May 2025 | **Revised:** 22 July 2025 | **Accepted:** 2 August 2025

Funding: This work was supported by a French government grant managed by the Agence Nationale de la Recherche under the France 2030 program, reference "ANR-22-PEFT-0008" - 203831.

ABSTRACT

In wireless communications, estimation of channels in OFDM systems spans frequency and time, which relies on sparse collections of pilot data, posing an ill-posed inverse problem. Moreover, deep learning estimators require large amounts of training data, computational resources, and true channels to produce accurate channel estimates, which are not realistic. To address this, a convolutional neural tangent kernel (CNTK) is derived from an infinitely wide convolutional network whose training dynamics can be expressed by a closed-form equation. This CNTK is used to impute the target matrix and estimate the missing channel response using only the known values available at pilot locations. This is a promising solution for channel estimation that does not require a large training set. Numerical results on realistic channel datasets demonstrate that our strategy accurately estimates the channels without a large dataset and significantly outperforms deep learning methods in terms of speed, accuracy, and computational resources.

1 | Introduction

Orthogonal frequency-division multiplexing (OFDM) is a modulation technique that has been extensively adopted in communication systems because it enables high data rate transmission while maintaining excellent bandwidth efficiency. OFDM divides the available spectrum into a number of overlapping but orthogonal narrowband subchannels, and hence converts a frequency-selective channel into a non-frequency-selective channel [1]. As a result, OFDM has started to be standardised as a key physical-layer technique in wireless LAN standards such as the American IEEE802.11a and the European equivalent HIPERLAN/2, and many commercial systems [2].

Deep learning (DL) has recently emerged as a powerful tool in communication systems, and Various DL approaches have been introduced to enhance the performance of traditional algorithms, particularly in tasks such as modulation recognition [3], signal detection [4] and channel estimation [5–8]. Notably, several studies have explored the use of deep convolutional neural networks (CNNs) for channel estimation [9–12], where both the input data and estimated channel responses are represented as images. In reference [8], a lightweight CNN was first used to extract relevant features from the pilot data. These extracted features were then passed through a transformer network to refine the channel estimation. Similarly, in reference [9], a CNN was used to estimate the channel by feeding the real and imaginary parts of least square (LS) estimated channels as input image, while training data was generated using the 5G New Radio Tapped Delay Line channel model. In references [12–14], conditional generative adversarial networks (cGAN)s were employed to estimate the channels as images. In reference [12], the input of the network was the quantised received signal, and the output was the estimated channel. In references [13, 14],

authors estimated the channels by cGANs while using the full channel image as groundtruth. Similarly, in reference [10] super resolution convolutional neural networks (SRCNN) model was proposed, taking a 2D noisy image as input representing the channel response at pilot locations, and producing the estimated channel as output. In all these works [8, 12–14], the channel response was estimated by training CNN models that use the simulated true/groundtruth channels to compute the network's training loss, which is not available in reality and requires large training time. Motivated by the above mentioned works, in this work, a lattice type pilot pattern is considered, and the time–frequency grid of fast fading communication channel is regarded as a three-dimensional (3D) image. The proposed contribution is to train an infinite width neural network [15, 16] for matrix imputation, which does not require a large dataset or true channels and significantly less time for channel estimation. The objective is to estimate the unknown values of the channel response while using only the known values at the pilot locations.

To the best of our knowledge, this is the first work exploring the concept of infinite width convolutional network for channel estimation. Several recent research have shown that wider neural networks provide notable advantages in terms of generalisation, classification accuracy and feature learning efficiency [17–19]. Interestingly, artificial neural networks (ANN)s behave like Gaussian processes as the network width tends to infinity [20, 21], which directly links to kernel methods. The proposed method (noted as channel-CNTK - channel convolutional neural tangent kernel) extends this idea further by using a convolutional neural tangent kernel (CNTK) [15] to impute the sparse time–frequency channel response image. The contributions of this letter are summarised below:

- To estimate the unknown values of the channel response from only the channel response at pilot positions, an infinitely wide CNN for kernel regression is employed.
- Numerical results show that our approach outperforms other deep learning approaches when large training sets or true channels are unavailable for training.

2 | Background

If the time–frequency grid is discretised for channel estimation as a grid of points, the prediction of channel response by interpolating data from existing channel response at pilot locations can be considered as a matrix imputation problem, where the missing matrix element values (the grid points) are estimated from the observed entries (pilot points). A number of learning-based methods for imputing missing values rely on supervised algorithms that use datasets with complete observations to find connections between missing and available data [22, 23]. Authors of references [15, 24, 25] introduced a method for regression, shape completion, and reconstruction using incomplete data by infinite width neural networks. Radhakrishnan et al. in reference [25] showed drug response prediction, and the authors of reference [15] showed regression by infinite neural networks. Interestingly, in reference [24], authors tackled reconstruction and unsupervised shape completion using sparse scanned data.

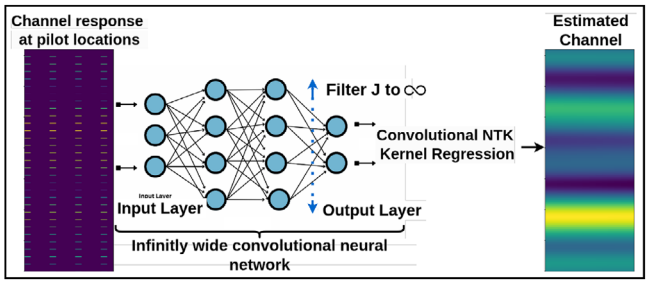

FIGURE 1 | An overview of the proposed Channel-CNTK method, which accurately and quickly estimates channel response by exploiting the width limits of neural networks. The main contribution lies in the computation of kernels.

In that work, the solution depends on a deep prior inspired by the neural tangent kernel (NTK).

3 | Methodology

3.1 | Inverse Problem Formulation

In this work, the goal is to estimate the channel time–frequency response $H \in \mathbb{C}^{M\times N}$ within an OFDM frame, where M and N represent the subcarriers and time slots (each time slot contains an OFDM symbol), respectively. The estimation relies solely on sparsely distributed pilot symbols P, transmitted at specific subcarriers and time slots. Each channel response is $H_{m,n} \in \mathbb{C}^{M\times N}$, located at coordinates (m, n) within the grid, where $m \in \{0, \dots, M-1\}$ and $n \in \{0, \dots, N-1\}$. The task of estimating the channel response at $H_{m,n}$ from these limited pilot observations P can be viewed as an inverse problem. The objective of channel estimation is to learn a function $\mathrm{f}_\theta : \mathbb{C}^P \to \mathbb{C}^{M\times N}$ capable of predicting the channel response at all subcarrier-time slot locations where pilots are not available. Here, θ denotes the parameters of the function f, and P represents the number of pilot observations. In this work, f is the infinite width network for kernel regression. A high level overview of the proposed Channel-CNTK method is shown in Figure 1.

3.2 | Neural Tangent Kernels

A neural network (NN) is a function $f_\omega(x)$ given by:

$$f_\omega(x) = \gamma_{\omega_L}\phi(\gamma_{\omega_{L-1}}\phi(\gamma_{\omega_{L-2}}\phi(\dots(\gamma_{\omega_1}(x))\dots)), \tag{1}$$

where x is the input and $\gamma_\omega \in \{1, \dots, L\}$, are the functions of the layer, followed by a nonlinear function known as activation ϕ., which According to reference [26], the training of infinite width networks can be described by a kernel function. Thus, with a kernel known as the NTK, solving kernel regression is similar to training infinite width networks [26]. We provide the CNTK formulation using the equation in section 4 in reference [15]. The CNTK tensor $\Theta \in \mathbb{R}^{M\times N\times M\times N}$ was defined and used in a regression setting. In this case, the CNTK is a 4-dimensional tensor where the inputs are coordinate features of the target matrix.

Definition (Neural Tangent Kernel). Let $f_\omega(x) : \mathbb{R}^P \to \mathbb{R}$ denote a neural network with initial parameters w_0. The **NTK,** $K : \mathbb{R}^{\mathbb{d}} \times \mathbb{R}^{\mathbb{d}} \to \mathbb{R}$ is a positive semi-definite function given by:

$$K(x, x') = \langle \nabla_\omega f_{\omega_0}(x), \nabla_\omega f_{\omega_0}(x') \rangle, \tag{2}$$

where $\nabla_\omega f_{\omega_0}(x)$ is the gradient and $w_0 \in \mathbb{R}^P$ denotes the parameters at initialisation.

Likewise, for convolutional networks, different number of layers, kernel size, and convolution, skip connections of neural networks are used to construct the CNTK (more details can be found in reference [15]).

Definition (Convolutional NTK). Let $\gamma_A(W)$ represent an L layer convolutional network which maps the feature of $A \in \mathbb{R}^{C\times M\times N}$ to the target matrix $Y \in \mathbb{R}^{M\times N}$, when the number of convolutional filter approaches infinity, the CNTK, $K : \mathbb{R}^{\mathbb{d}} \times \mathbb{R}^{\mathbb{d}} \to \mathbb{R}$ of γ_A is a positive semi-definite kernel given by:

$$K(X_{m,n}, X_{m',n'}) = [\Theta^L(A, A')]_{m,n,m',n'}, \tag{3}$$

where, $X_{m,n}, X_{m',n'} \in \mathbb{R}^{M\times N}$ is the indicator matrices at coordinate (m, n) and (m', n') and Θ^L is the tensor at the final layer L. As derived in reference [15], if $f(X) = X^T\gamma_A(W)$ for $X \in \mathbb{R}^{M\times N}$ the kernel at the indicator matrices is given by:

$$\begin{aligned} K(X_{m,n}, X_{m',n'}) &= \frac{\delta(\gamma_A(W)_{m,n})}{\delta W_{\alpha,\beta}} \frac{\delta(\gamma_A(W)_{m',n'})}{\delta W_{\alpha,\beta}} \\ &= [\Theta^L(A, A')]_{m,n,m',n'} \end{aligned} \tag{4}$$

where $f(X)$ is a function of the infinite width CNN's output and input with respect to $W_{\alpha,\beta}$, which are the indexed parameters. $A \in \mathbb{C}^{C\times M\times N}$ is a prior tensor, where C is the number of channels and W are the weights. In this study, we initialised A using a *prior*, which resembles semi-supervised learning and captures the relationship among the coordinates within the target matrix. In this work, in the context of channel estimation, similar to local image prior (LIP) used in reference [27], the sparse H^p_{LS} image, is used to initialise A for the CNN model.

3.3 | CNN Model Architecture

When the network width tends to infinity, the CNTK is derived using a CNN. The proposed CNN model is modified from reference [28] and has six convolutional layers, each using a kernel size of (3, 3), followed by *LeakyReLU* activation, where 0.05 and 1 are the slopes, providing nonlinearity while maintaining kernel tractability. Transposed convolution, upsampling or trilinear upsampling are employed to change the spatial dimensions of the feature maps. The hyperparameters of the CNN model, such as the number of layers, kernel size, and activation function, were selected based on prior work [23] by empirical validation to balance performance and computational efficiency.

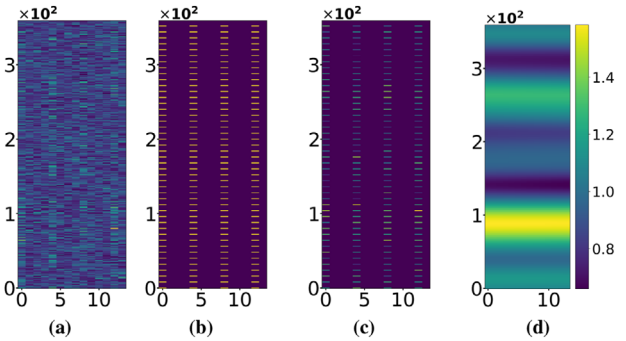

FIGURE 2 | Example of a sample of channel time–frequency response images. (a) The received signal Y, (b) transmitted symbols X, (c) channel response at pilot locations H^p_{LS} (the target matrix which we want to impute) and (d) perfect channel estimate H_{perf}, by MATLAB 5G toolbox [29].

3.4 | Image of Channel Response Time–Frequency Grid

In this study, we focus on a link between a pair of Tx and Rx antennas in a device-to-device fast-moving scenario. The channel time–frequency response matrix, denoted as H (of size $M \times N$) characterises the link between the transmitter and receiver using complex values. This can be considered as a 3Dimage, that is, $H \in \mathbb{C}^{C_H\times M\times N}$, where C_H, M and N are the channel, height, and width of the channel image, respectively. In an OFDM system, the received signal $Y_{m,n}$ after removal of cyclic prefix and discrete Fourier transform performed at the mth subcarrier and nth time slot can be expressed by Equation (5):

$$Y_{m,n} = X_{m,n}H_{m,n} + Z_{m,n}, \tag{5}$$

where $Y, X, H, Z \in \mathbb{C}^{M\times N}$ and $X_{m,n}$, $H_{m,n}, Z_{m,n}$ represent the known transmitted symbol, channel response and additive white Gaussian noise (AWGN) with zero mean and variance σ^2_w at the mth subcarrier and the nth time slots, respectively. Following the least square (LS) estimation specifically with fading, the estimate of the channel H^p_{LS} at pilot positions can be expressed as:

$$H^p_{\text{LS}} = Y/X. \tag{6}$$

A visual representation of normalised image for a sample of received signal Y, estimated channel response at pilot locations H^p_{LS}, (i.e., the target matrix which we want to impute) and perfect estimation of the channel H_{perf} on time–frequency grid with $M =$ 14 time slots and $N = 360$ subcarriers (based on 5G New Radio [5G-NR] 3GPP standard [29]) is shown in Figure 2.

3.5 | Estimating the Channel With CNTK

In this case, the sparse channel response H^p_{LS} (see Figure 2c) has dimensions $M \times N$. The sub-tensors $H^p_{\text{LS}_{\text{sub}}}$ from H^p_{LS}(in the following sections, $M = 12, N = 14$ is chosen, sub-tensor details are given in section 4.2) and the CNTK $K \in \mathbb{C}^{M\times N\times M\times N}$. Let $\mathbf{X}'$ be the set of available channel response locations and $\mathbf{y}$ the corresponding channel response; $\mathbf{X}$ is the location of an unobserved point where the prediction of the channel response

is made. The predicted value $\hat{H}(\mathbf{x})$ is given by:

$$\hat{H}(\mathbf{x}) = K(\mathbf{X}, \mathbf{X}')^{T}.K(\mathbf{X}, \mathbf{X})^{-1}.\mathbf{y} \tag{7}$$

where $K(\mathbf{X}, \mathbf{X}')$ is the CNTK evaluated between the training data and the predicted location. $K(\mathbf{X}, \mathbf{X})$ and X is the CNTK evaluated using X as the training data.

4 | Numerical Results

In this section, channel-CNTK approach is evaluated with different pilot density using traditional interpolation techniques and neural network models, namely, ANN and cGAN.

4.1 | Evaluation Metric

The normalised mean squared error (NMSE) is utilised to calculate the difference between the estimated matrix $\hat{H}$ and the true channel matrix H, which is expressed as:

$$\text{NMSE} = 10\log_{10}\left\{\mathbb{E}\left[\frac{\|H-\hat{H}\|^2}{\|H\|^2}\right]\right\} \tag{8}$$

where $||\cdot||$ denotes the matrix norm calculation and E obtains values of expectation. And $10\log_{10}\{\cdot\}$ is calculated to obtain NMSE values in decibels.

4.2 | Implementation Details and Datasets

As shown in reference [9], the dataset is generated using the MATLAB 5G Toolbox [29], employing the 3GPP Tapped Delay Line (TDL-A) channel model for realistic channel modelling and pilot transmission. MATLAB is used only to create the dataset and was not used elsewhere for the DL-based channel estimation. Pilot symbols are placed in every fourth subcarrier and every second OFDM symbol, which can be adjusted to increase the density of pilots to 24, 16, and 12 pilots per resource block across the time–frequency grid. The number of resource blocks in our experiments was 30, but it can be 15, 30, etc. The target is to estimate the channel from each sparse H^{p}_{LS} image. The $H^{p}_{\text{LS}} \in \mathbb{C}^{C_H \times M \times N}$, where $M = 360$, $N = 14$ and $C_H = 1$. Given the large size of the tensors, H^{p}_{LS} is split into 30 equal parts as $H^{p}_{\text{LS}_{\text{sub}}}$ with size $12 \times 14 \times 1$ for the Channel-CNTK imputation process. After imputation, each sub-tensor $H^{p}_{\text{LS}_{\text{sub}}}$ were stitched together to form the full estimated channel. For each image of H^{p}_{LS}, the best-performing configuration of the proposed CNTK model—using 720 training samples and the remaining pixels as test samples, the baseline deep learning (DL) models and traditional interpolation methods were also trained using 720 known (training) pixels and evaluated on the remaining unknown (test) pixels. That is approximately only 7% of the total pixels available in a $360 \times 14 \times 1$ image. Both ANN and cGAN generator models were trained with 0.001 learning rate, ADAM optimiser, and mean squared error (MSE) loss. Deep learning model (cGAN and ANN) architecture and training parameters are given in references [30–33]. For the channel models of TDL-A (with short delay spread) with carrier frequency of 2.1 GHz, a Doppler shift of approximately 175 Hz, and user equipment (UE) speed of 50 km/h, are considered. For each pilot arrangement (12, 16, and 24), the dataset contains 100 samples.

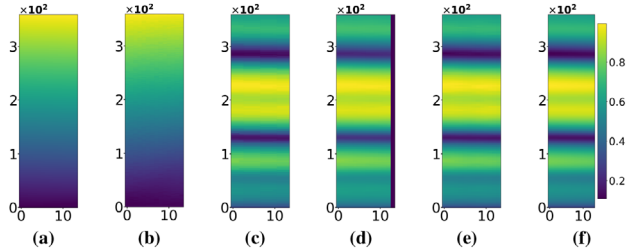

FIGURE 3 | Estimated channels by different methods: (a) and (b) show DNN and cGAN estimations, (c) and (d) present fuzzy and degraded estimates from KNN and linear methods, (e) showcases our Channel-CNTK estimation and (f) shows the perfect channel estimation from MATLAB for comparison, not used for estimation.

4.3 | Visual Analysis

The estimated channels by hannel-CNTK and comparison with other approaches are shown in pseudo-colour images in Figure 3. The results of the DL models, in Figure 3a (DNN) and 3(b) (cGAN), struggle to generalise and learn the channel features. This is because there are few training samples (24) per resource block that are insufficient to train a DNN or a cGAN. A robust cGAN with two neural network models – generator (G) and discriminator (D), having a large number of parameters, trained on limited features and only 7% of total pixels in each H^{P}_{LS}, is not reliable. This leads to poor accuracy and generalisation performance of the cGAN and ANN models. In Figure 3c, KNN estimates the reference (groundtruth) channel response (Figure 3f). Upon closer inspection (zooming), one can clearly see notable artefacts and fuzzy distortions within the interpolated areas. Linear interpolation in Figure 3d fails to accurately capture the channel characteristics due to its inherent linear assumptions (the non-interpolated region can be seen on the right side of the image). In contrast, in Figure 3e, the proposed CNTK-based method shows remarkable performance, very closely approximating the reference perfect channel response (Figure 3f). The results confirm that channel-CNTK effectively captures the spatial features of the channel, outperforming deep learning and traditional interpolation methods on a limited number of training data points in an image.

4.4 | Quantitative Analysis and Impact of Pilot Density

To further validate the results of visual inspection, we compare the impact of various SNR conditions on channel estimation accuracy. Figure 4 shows that CNTK consistently outperforms deep learning models and traditional interpolation methods across SNRs ranging from −10 dB to −30 dB, with channel estimation accuracy improving as the number of pilot symbols increases. The cGAN and DNN models exhibit poor performance due to the insufficient number of training points in each sample of H^{p}_{LS}, limiting their ability to learn from the channel features. A similar trend is observed for linear interpolation. KNN and CNTK-12 also show performance degradation at high SNRs, attributed to the limited training samples. In contrast, channel-CNTK consistently achieves the best results across all SNR values,

TABLE 1 | Comparison of methods in terms of memory usage and speed.

Method	Memory usage	Training/image	Inference/image
DNN	1.2 GB RAM + 320 MB VRAM	40 s	1 s
cGAN	10 GB RAM + 10 GB VRAM	46 min	1 s
Linear	1.2 GB RAM	—	1.85×10^{-2}s
KNN	1.2 GB RAM	—	1.846×10^{-2}s
Channel-CNTK	1.2 GB RAM	—	1.48×10^{-4}s

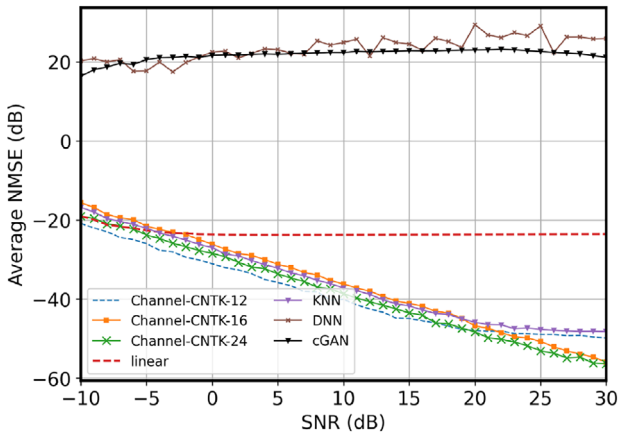

FIGURE 4 | Performance of different methods with varying SNRs.

even with only 24 pilots per resource block, and demonstrates robustness at low SNRs compared to other methods.

This can be attributed to the ability of the infinite width network to extract feature maps from the structure of the data and provide accurate estimations of the CNTK, therefore mitigating the influence of noise on kernel regression and proving the reliability of our CNTK approach.

4.5 | Comparative Analysis of Time Efficiency of the Algorithms

We examined the efficiency of the Channel-CNTK method in terms of memory consumption and the time required for inference. Our analysis shows that the cGAN and DNN approaches required approximately 46 minute during 500 epochs and 46 seconds during 1000 epochs on one image to train, respectively, on a machine with 32GB RAM and 32GB GPU memory (Intel core i7, NVIDIA ADA 3500). Both models required higher memory, specifically for DNN and cGAN, 320 MB VRAM and 10GB VRAM for training. In contrast, as shown in Table 1, to train and impute the channel matrix, channel-CNTK method took only 1.48×10^{-4} seconds to impute one complete channel image. The CNTK computation time was 1.2×10^{-1} seconds.

5 | Conclusion

In this letter, we propose a fast channel estimation framework using an infinitely wide CNN. In contrast to alternative channel estimate methods, this method remarkably reduces the time required and computational resources (memory, machine configuration, etc.). Moreover, a large dataset of any groundtruth channels generated from a simulator is not required for channel-CNTK, as required for training deep learning models. From only a few pilots in an image, the proposed CNTK-based method can estimate the channel response fast with accuracy in under 1.48×10^{-4} s. In conclusion, the Channel-CNTK is a solution for channel estimation, offering a balance between efficiency, computational resources, and accuracy when no large dataset or groundtruth is available. channel-CNTK can be extended to uplink, downlink or massive MIMO channel estimation, with further refinements and validations planned to enhance its robustness and applicability in the next work.

Author Contributions

Mohammed Mallik: investigation, methodology, data preparation, software, validation, writing – original draft. **Guillaume Villemaud:** funding acquisition, writing – review.

Acknowledgements

This work was supported by a French government grant managed by the Agence Nationale de la Recherche under the France 2030 program, reference "ANR-22-PEFT-0008" - 203831.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. Y. G. Li, J. H. Winters, and N. R. Sollenberger, "MIMO-OFDM for Wireless Communications: Signal Detection With Enhanced Channel Estimation," *IEEE Transactions on Communications* 50, no. 9 (2002): 1471–1477.

2. "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," in *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp.1–4379, (2026) Feb. 2021, https://doi.org/10.1109/IEEESTD.2021.9363693.

3. T. O'shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking* 3, no. 4 (2017): 563–575.

4. N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO Detection," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)* (IEEE, 2017), 1–5.

5. J. Li, Z. Zhang, Y. Wang, B. He, W. Zheng, and M. Li, "Deep Learning-Assisted OFDM Channel Estimation and Signal Detection Technology," *IEEE Communications Letters* 27, no. 5 (2023): 1347–1351.

6. P. E. G. Silva, J. M. Moualeu, P. H. Nardelli, Y. Li, and R. A. A. de Souza, "An Efficient Machine Learning-Based Channel Prediction Technique for OFDM Sub-Bands," in *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)* (2024), 1–5.

7. C. Li, T. Zhang and S. Liu, "CrossNet: Joint Channel Estimation and Localization in Deep Learning Method," in IEEE Communications Letters, vol. 29, no. 4, pp. 789-793, April 2025, https://doi.org/10.1109/LCOMM.2025.3543579.

8. J. Li and Q. Peng, "Lightweight Channel Estimation Networks for OFDM Systems," *IEEE Wireless Communications Letters* 11, no. 10 (2022): 2066–2070.

9. M. H. Ahmed, T. M. Jamel, and H. F. Khazaal, "5G channel estimation for SISO-OFDM communication systems using deep learning," in AIP Conference Proceedings, Vol. 3002, No. 1 (AIP Publishing LLC, 2024, June), 020004.

10. M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep Learning-Based Channel Estimation," *IEEE Communications Letters* 23, no. 4 (2019): 652–655.

11. A. K. Gizzini, M. Chafii, A. Nimr, R. M. Shubair, and G. Fettweis, "CNN Aided Weighted Interpolation for Channel Estimation in Vehicular Communications," *IEEE Transactions on Vehicular Technology* 70, no. 12 (2021): 12 796–12 811.

12. Y. Dong, H. Wang, and Y.-D. Yao, "Channel Estimation for One-Bit Multiuser Massive MIMO Using Conditional GAN," *IEEE Communications Letters* 25, no. 3 (2020): 854–858.

13. Y. Guo, Z. Qin, X. Tao, and O. A. Dobre, "Federated Generative-Adversarial-Network-Enabled Channel Estimation," *Intelligent Computing* 3 (2024): 0066, https://spj.science.org/doi/abs/10.34133/icomputing.0066.

14. M. Ye, C. Pan, Y. Xu, M. Jiang, X. Liang, and C. Li, "Channel Estimation Based on An Improved Conditional GAN for MIMO-OFDM Systems," in 2023 IEEE/CIC International Conference on Communications in China (ICCC) (IEEE, 2023, August), 1-6.

15. S. Arora, S. S Du, W. Hu, Z. Li, R. R Salakhutdinov, and R. Wang, "On exact computation with an infinitely wide neural net," in Advances in Neural Information Processing Systems (2019), 8141–8150.

16. C. K. I. Williams, "Computing with infinite networks," in *Advances in neural information processing systems* (1997), 295–301.

17. J. Lee, L. Xiao, S. Schoenholz, et al., "Wide neural networks of any depth evolve as linear models under gradient descent," in: *Advances in Neural Information Processing Systems* (2019), 8572–8583.

18. R. Novak, J. Sohl-Dickstein, and S. S. Schoenholz, "Fast Finite Width Neural Tangent Kernel," in *Proceedings of the 39th International Conference on Machine Learning*, eds. K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, 162, no. 17–23 (PMLR, 2022), 17 018–17 044, https://proceedings.mlr.press/v162/novak22a.html.

19. V. Nagarajan and J. Z. Kolter, "Uniform convergence may be unable to explain generalization in deep learning," in: *Advances in Neural Information Processing Systems* (2019), 11611–11622.

20. R. Novak, L. Xiao, J. Lee, et al., "Bayesian Deep Convolutional Networks With Many Channels are Gaussian Processes," (2020), https://arxiv.org/abs/1810.05148.

21. A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani, "Gaussian Process Behaviour in Wide Deep Neural Networks," preprint, *arXiv*:1804.11271 (2018).

22. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image Inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00 (ACM Press/Addison-Wesley Publishing Co., 2000), 417–424, https://doi.org/10.1145/344779.344972.

23. M. Mallik, B. Allaert, E. Egea-Lopez, D. P. Gaillot, J. Wiart, and L. Clavier, "Infinite Limits of Convolutional Neural Network for Urban Electromagnetic Field Exposure Reconstruction," *IEEE Access* 12 (2024): 49 476–49 488.

24. L. Chu, H. Pan, and W. Wang, "Unsupervised Shape Completion via Deep Prior in the Neural Tangent Kernel Perspective," 40, no. 3 (ACM New York, 2021), 1–17.

25. A. Radhakrishnan, G. Stefanakis, M. Belkin, and C. Uhler, "Simple, Fast, and Flexible Framework for Matrix Completion With Infinite Width Neural Networks," *CoRR* abs/2108.00131 (2021), https://arxiv.org/abs/2108.00131.

26. A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Proc. NeurIPS* (2018), 8571–8580.

27. M. Mallik, D. P. Gaillot, and L. Clavier, "GLIP: Electromagnetic Field Exposure Map Completion by Deep Generative Networks," in *2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* (IEEE, 2024), 1–5.

28. C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-Resolution Image Inpainting Using Multi-Scale Neural Patch Synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2017), 6721–6729.

29. T. M. Inc, 5G Toolbox (r2024b) (2024), https://www.mathworks.com.

30. S. Rehman and M. Mohandes, "Artificial Neural Network Estimation of Global Solar Radiation Using Air Temperature and Relative Humidity," *Energy Policy* 36, no. 2 (2008): 571–576.

31. M. Abadi, P. Barham, J. Chen, et al., "TensorFlow: a system for Large-Scale machine learning," in *12th USENIX symposium on operating systems design and implementation* (OSDI 16, 2016), 265-283.

32. J. Oskarsson Probabilistic regression using conditional generative adversarial networks (2020).

33. P. Virtanen, R. Gommers, T. E. Oliphant, et al., "Scipy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods* 17, no. 3 (2020): 261–272.