

# Deep Learning for MPC Extraction and False Detection

Howard Dai<sup>a</sup>, Jack Chuang<sup>b</sup>, Jian Wang<sup>b</sup>, Samuel Berweger<sup>c</sup>, and David Griffith<sup>b</sup>

<sup>a</sup>Department of Computer Science, Yale University, USA

<sup>b</sup>Communications Technology Laboratory, NIST, Gaithersburg, Maryland, USA

<sup>c</sup>Communications Technology Laboratory, NIST, Boulder, Colorado, USA

## ABSTRACT

Multipath component (MPC) extraction is critical for channel modeling and for integrated sensing and communications (ISAC). The super-resolution algorithm known as CLEAN-Space-Alternating Generalized Expectation (CLEAN-SAGE) is widely used for MPC extraction. Because CLEAN-SAGE is based on maximum likelihood estimation (MLE), it can experience performance degradation and false detection events when a mismatch occurs between the theoretical model and the received signal. Moreover, the complexity of CLEAN-SAGE makes it challenging to support real-time or near-real-time applications because the algorithm iteratively extracts parameters of MPCs with gradient search. This paper proposes two machine learning (ML)-based solutions to address these issues: a classification model for false signal detection and regression models for direct MPC parameter extraction. The classification model uses a convolutional neural network (CNN) to predict true or false detection from the beamformed signals, which can be used alongside CLEAN-SAGE to reduce false detection rates. Our model achieves greater than 90 % classification accuracy on simulated data and a smaller set of real-world dataset collected by NIST's 141 GHz phased array channel sounder. We developed two regression models using Redmon's "You Only Look Once" YOLO model to focus on reducing estimation errors, and a CNN-based autoencoder to focus on processing speed. Both models use a heatmap to directly predict the number of MPCs, the range, and the azimuth angle of each MPC. We leverage the NIST Wi-Fi ISAC simulation bench to generate heatmap training data in the signal-to-noise ratio (SNR) range of  $-10$  dB to  $25$  dB before pulse compression. Our models achieve accuracy comparable to CLEAN-SAGE and a false detection rate less than 0.5 % in multi-target scenarios. The results also demonstrate that all our regression models are faster than CLEAN-SAGE.

**Keywords:** Deep learning, NextG, Joint communications and sensing, Channel Sounding.

## 1. INTRODUCTION

Integrated Sensing and Communication and sensing has drawn growing interest in recent years from both the research community and 5G standards. Sensing and communication can share common hardware modules and signal processing modules and use the same waveform to reduce operation costs and achieve spectrum efficiency. With ISAC, wireless devices can be transformed from mere communication tools into sensors that can perceive their surroundings to enable many emerging smart applications, such as health monitoring, smart home, smart transportation, virtual reality, and more.

To achieve robust sensing outcomes, the high accuracy Multipath Components parameter estimation of delay and Angle-of-Arrival(AoA) are required, and these parameters are also critical for the channel modeling community. The CLEAN-SAGE super-resolution algorithm is frequently applied for these applications to accomplish the tasks<sup>1,2</sup>. However, research has revealed that the model mismatch between the received signals and the theoretical system model will limit the accuracy and reliability of the estimation and potentially create false detection<sup>3,4</sup>. Because of the difficulty in mathematically modeling the characteristics of the false detection, the false detection will usually propagate into channel estimation outputs and impact the quality of sensing and channel modeling. In<sup>5</sup>, authors proposed MPC rejection criteria to reduce the number of false detections. These criteria are based on the power of the current MPC, extracted MPCs should stay within the Field-of-View (FoV) of the system, and the minimum distance between the current MPC and the MPCs extracted in spherical. Nevertheless, the paper did not provide a detailed architecture to implement these criteria and did not have performance validation in simulated or real data.

Deep Learning (DL) has gained interest in the application of channel estimation.<sup>6</sup> The machine learning based methods provide alternative solutions to existing methods based on traditional digital signal processing. In,<sup>7</sup> a Deep Neural Network (DNN) is used to estimate the channel information matrix of a Multi-Input and Multi-Output (MIMO) communication system under frequency selective fading. Two Convolutional Neural Network (CNN) based architectures are also proposed to estimate MIMO channel information matrix under the Doppler and frequency selective fading,<sup>8,9</sup> A CNN-based method predicting delay and Doppler from complex baseband samples for the ISAC system is also developed in.<sup>10</sup> Machine learning has also been used to assist channel modeling efforts to predict delay and angle spread.<sup>11</sup> All the above research demonstrates the potential of machine learning to resolve channel estimation and channel modeling problems with comparable results to traditional signal processing. However, none of these works discuss false detection from the system, and they do not simultaneously predict delay and AoA from the beamforming heatmaps.

To address the above concerns, we proposed a DL model to resolve false detection and two DL models for MPC parameters estimation. The classification model uses a CNN to predict truth or false detection given an input signal, which can be used alongside CLEAN-SAGE to reduce false detections. To overcome the small training data set from the real system, we use NIST's 141 GHz phased array antennas to collect system uncertainty statistics and integrate them into our simulation bench to generate large quantities of diverse training data. The first regression model uses multiple convolutional autoencoders from CNN with several fully connected linear layers to directly predict the numbers of targets and X-Y coordinates of each target from the heatmap. This model focuses on processing speed and shows significant speed improvement compared with CLEAN-SAGE with acceptable performance degradation on coordinates estimation. The second model is modified from YOLO and focuses on estimation performance. It demonstrates comparable estimation performance compared with CLEAN-SAGE while still maintaining an advantage in processing speed. We leverage the NIST Wi-Fi ISAC simulation bench to generate large amounts of heatmap training data.<sup>12</sup>

To summarize, the key contributions of our work are as follows:

- Developed a CNN-based classification model to identify false detection from CLEAN-SAGE that achieved greater than 90 % accuracy on both simulated and real-world data.
- Two regression models to extract MPC parameters have been developed. One is focusing on processing speed and the other is focusing on estimation performance. The AoA and delay of one or multiple MPCs from the beamforming heatmap have been estimated and then mapped to X and Y in meters in the Cartesian coordinate.

The remainder of the paper is organized as follows: Section 2 introduces the classification model and training data preparation. We also introduce the NIST 141 GHz phased array channel sounder used for concept validation. Section 3 describes the regression model and introduction of the NIST WiFi simulation bench used for generating training data. Section 4 provides the performance evaluation results for both classification and regression models. Section 5 concludes our paper.

## 2. SYSTEM OVERVIEW AND FALSE DETECTION

### 2.1 System Overview

The theoretical received signal from the  $m$ -th MPC at the  $i$ -th element of a phased array can be expressed as,

$$r(t) = A_m s(t - t_m) \exp(j \frac{2\pi}{\lambda} (x_i \cos(\theta_m) \cos(\phi_m) + y_i \cos(\theta_m) \sin(\phi_m) + z_i \sin(\theta_m))) \quad (1)$$

Where  $A_m$ ,  $t_m$ ,  $\theta_m$ , and  $\phi_m$  are complex amplitude, delay, elevation, and azimuth of the  $m$ -th MPC need to be estimated. The variables  $x_i$ ,  $y_i$ , and  $z_i$  are the Cartesian coordinates of  $i$ -th phase array element. The wavelength  $\lambda$  is 2.1mm. Because of time-varying hardware operation status and system jitters, the received signal in the real system can be rewritten as,

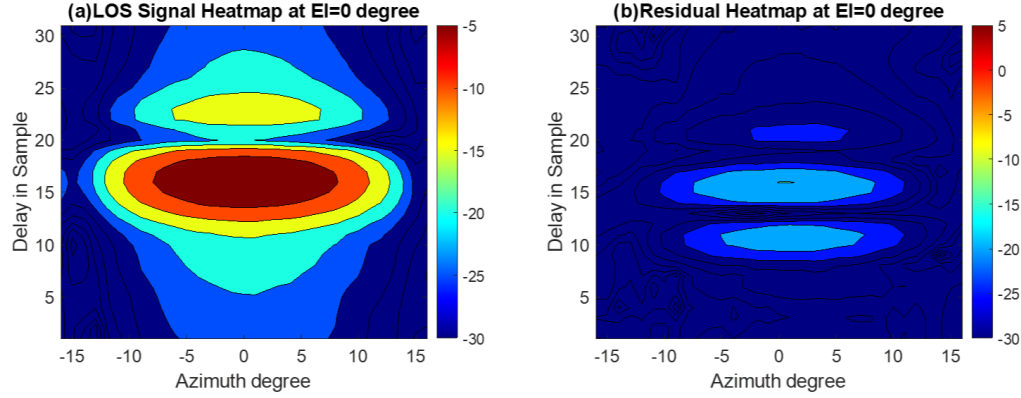


Figure 1. (a) Signal heatmap over the delay and azimuth domain (b) Residual heatmap over the delay and azimuth domain

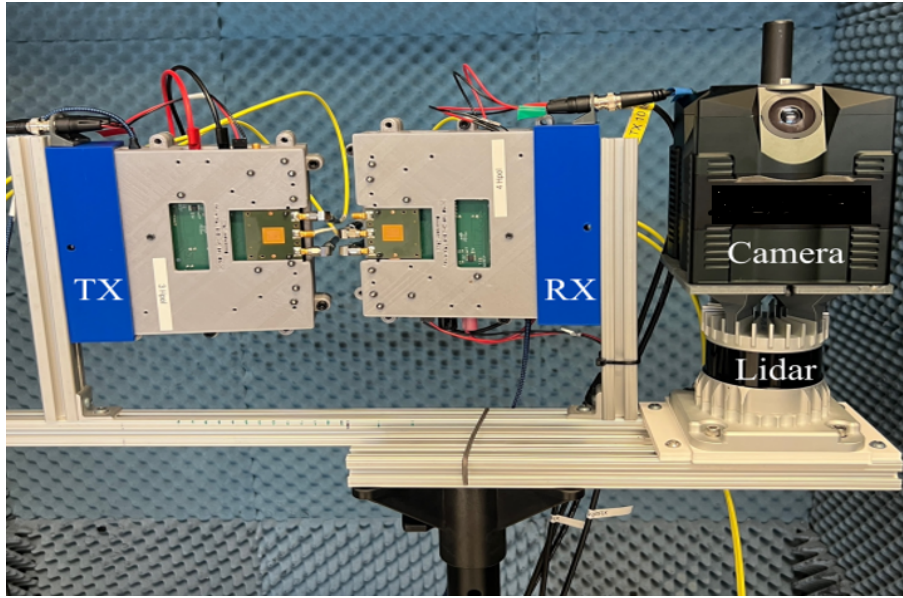


Figure 2. TX and RX boards mounted in a monostatic configuration together with the camera and Lidar

$$r(t) = A'_i A_m s(t - t_m - t'_i) \exp(\mu_i) \exp(j \frac{2\pi}{\lambda} (x_i \cos(\theta_m) \cos(\phi_m) + y_i \cos(\theta_m) \sin(\phi_m) + z_i \sin(\theta_m))) \quad (2)$$

Where  $A'_i$ ,  $t'_i$ , and  $\mu_i$  are random amplitude, time, and phase jitters from the system. The model mismatch between Equations- 1 and 2 could result in a strong residual in the expectation step of the CLEAN-SAGE and trigger the false detection. A visualization example of a signal and residual heatmap in the beamformed power-delay-profile (PDP) domain at an elevation equal to 0 degrees is shown in Figure 1(a) and Figure 1(b). We use the NIST 141 GHz phased array channel sounder to collect beamformed PDP and feed it to the CLEAN-SAGE algorithm to generate Figure 1. The residual after removing the estimated signal is 20 dB weaker than the signal. However, the power of the residual is still strong enough to trigger false detection.

Figure 2 is the photo of NIST channel sounder collecting heatmap in Figure 1. Our channel sounder is vertically polarized and can operate in either transmitter or receiver mode.<sup>13</sup> The 64 microstrip antennas are spaced roughly at half-wavelength: 1.2 mm in azimuth (horizontal) and 1.1 mm in elevation (vertical). The arrays can form a beam within a FoV that corresponds to the 3-dB beamwidth of the individual antennas, which is  $\pm 39$  degrees in azimuth and  $\pm 43$  degrees in elevation; the individual antennas have 3.5 dBi boresight gain.

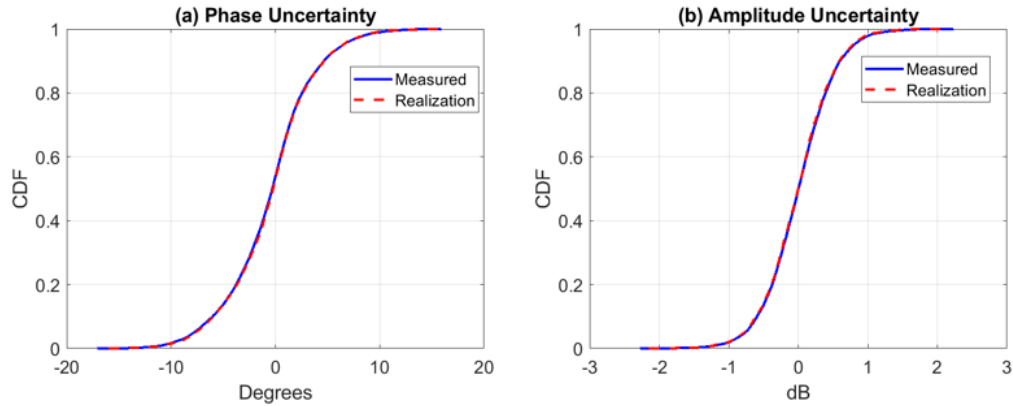


Figure 3. (a) Phased uncertainty: Measured vs. Realization (b) Amplitude uncertainty: Measured vs. Realization

## 2.2 AI on False Detection

CNN-based architectures excel at capturing spatial relationships and learning abstract shapes, and are most commonly applied to image-like data.<sup>14</sup> In this task, a channel impulse response is a 3-dimensional heatmap over azimuth, elevation, and delay, so our input data resembles a 3D image with low resolution. Thus, we treat this as image classification and proceed with a CNN-based architecture.

One of the challenges for deep learning models is requiring a large amount of training to perform well. This requirement necessitates our system to take large amounts of data in different environments which is not feasible. To overcome this problem, we extract our system's uncertainty statistics  $A'$ ,  $t'$ ,  $\mu$  and apply them into Equation-2 to generate simulated training data with behavior similar to the real system. To extract system uncertainty statistics, our sounder repeated the measurement 225 times in a controlled environment to obtain the empirical cumulative density functions (CDF) of the system uncertainty and then applied the inversion method to generate random variables for amplitude, time, and phase uncertainty.<sup>15</sup> Figure 3 shows the CDF of phase and amplitude uncertainty between measured and realization. The realization random variables passed the Kolmogorov-Smirnov (KS) test, ensuring they have the same distribution as measured data.

The system uncertainty parameters are injected into the simulation bench to generate beamformed PDP. There are 1 to 3 targets per simulation with an SNR range of 12 to 30 dB after pulse compression. We apply CLEAN-SAGE on each beamformed PDP and compile each individual detection into a dataset, labeling each detection as true or false.

By applying CLEAN-SAGE to 1000 simulated scenarios, we produce a simulated dataset of 4335 target detections (2416 real, 1919 false). To demonstrate the transferability from simulated to real system data, the NIST 141 GHz channel sounder was used to collect 49 scenarios under line of sight (LOS) with azimuth and elevation angles randomly picked between  $[-20\ 20]$  degrees. Applying CLEAN-SAGE to these scenarios produces an additional real-world test dataset containing 117 target detections (49 real, 68 false).

Figure 4 demonstrates our classification architecture. The classification model contains 5 3D-convolutional layers and 3 fully connected layers.<sup>16</sup> ReLU (Rectified Linear Unit), maximum pooling, and dropout are applied to each layer.<sup>17</sup> Training and evaluation details for the classification model will be presented in Section 4.

## 3. WIFI SENSING AND MPC EXTRACTION

### 3.1 WiFi Sensing Bench

The WiFi simulation bench, based on the IEEE 802.11ad standard, has been used to generate training and validation data for the regression model. We utilize control PHY (CPHY) packed and a channel estimation field (CEF) for sensing and generating heatmap. Our system model adopts a monostatic configuration in which both transmitter and receiver are deployed at the WiFi access point (AP).

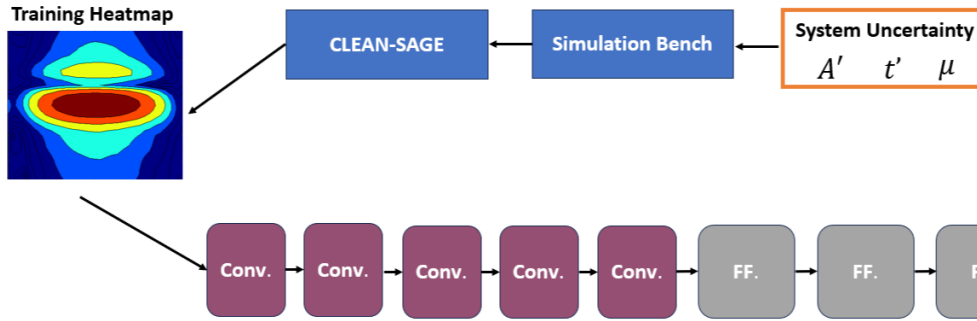


Figure 4. False detection architecture

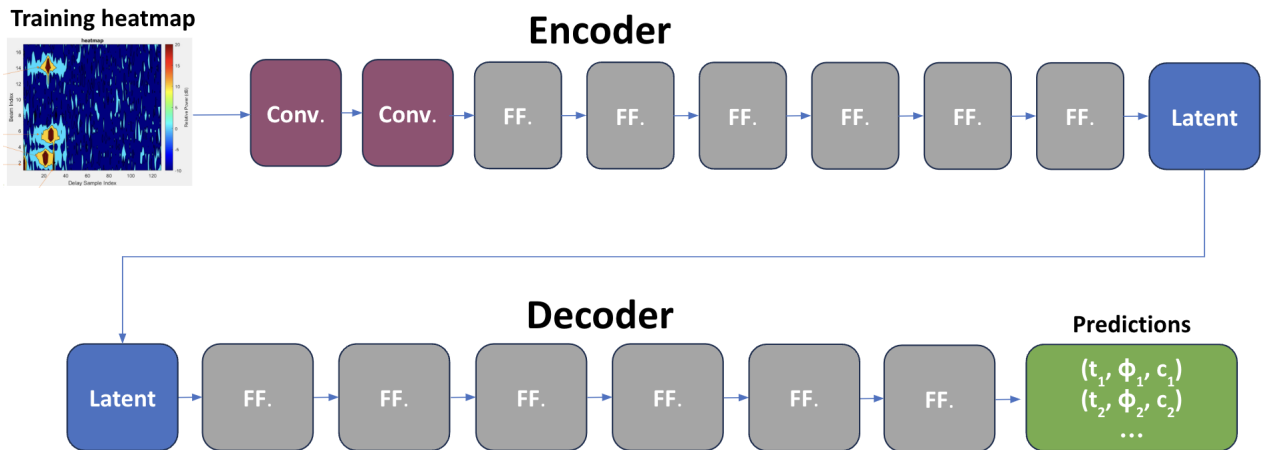


Figure 5. MPC Extraction architecture

A uniform linear array (ULA) with 16 elements forms 17 beams uniformly distributed over the azimuth range between  $[-45, 45]$  degrees. The complementary Golay sequences with a length of 128 are used for channel estimation, and the heatmap size will be a  $17 \times 128$  matrix. In the simulation, the one to eight targets with an SNR range of  $-10$  dB to  $25$  dB are randomly placed in the delay and azimuth domain, which correspond to  $xy$ -coordinates in a  $7 \times 7$  meters room.

### 3.2 AI on MPC Extraction

Contrasting CLEAN-SAGE, AI-based methods are less sensitive to noise; thus, using machine learning to directly predict MPC coordinates can naturally mitigate false detections when used on noisy real-world data. In addition, using oneshot ML-based methods can also provide advantages in terms of inference speed.

We implement and compare two machine learning methods: a general CNN regression model, and the pre-trained YOLO vision model developed by Redmon et al.<sup>18</sup>

**CNN** We implement a similar architecture as that described in Section 2.2, with a few key differences. Rather than binary classification, our model must output coordinate predictions for one or more targets. In addition, we use 2D convolutional layers, as the WiFi sensing dataset does not include information across varying elevations. We also utilize only two convolutional layers; while translational invariance is crucial in classification task, for MPC extraction, we emphasize positional information. Figure 5 depicts our regression architecture.

To capture a variable number of predictions, we format our output as fixed  $N \times 3$  array, representing  $N$  predictions  $(\hat{t}_i, \hat{\phi}_i, \hat{c}_i)$ , representing delay, azimuth, and confidence, respectively. Here,  $N$  must be an upper bound to the number of potential MPCs in any given heatmap; because our dataset includes examples ranging

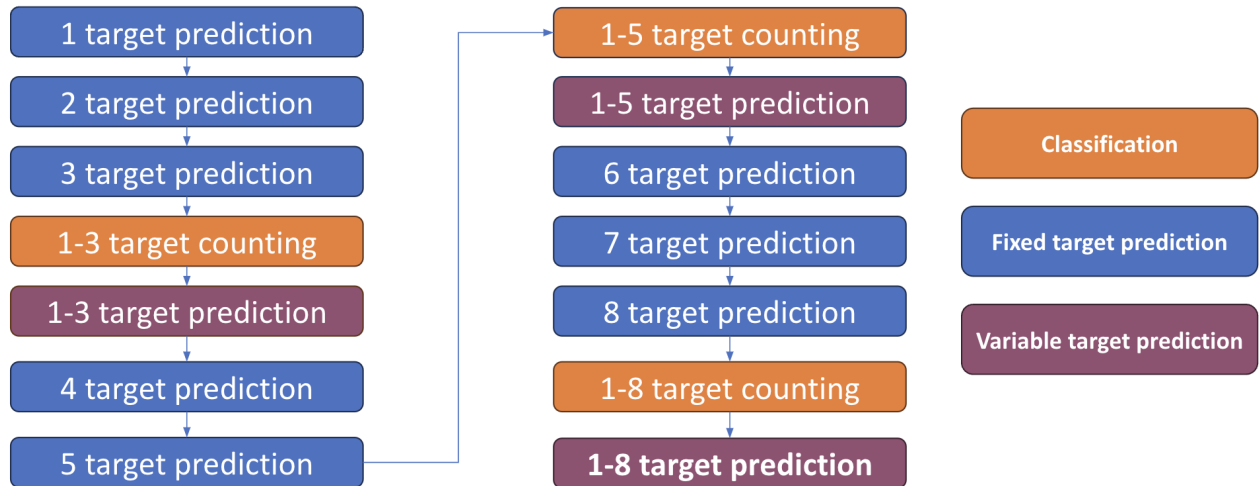


Figure 6. Sequence training overview

from 1-8 targets, we set  $N = 10$ . To compute loss for an example with  $n$  targets, we greedily pair ground-truth targets  $(t_j, \phi_j)$  with closest predictions  $(\hat{t}_{\psi(j)}, \hat{\phi}_{\psi(j)}, \hat{c}_{\psi(j)})$ . We then define the following MSE loss:

$$L_{\text{MSE}} = \sum_j^n A(t_j - \hat{t}_{\psi(j)})^2 + B(\phi_j - \hat{\phi}_{\psi(j)})^2 \quad (3)$$

We set  $A, B$  to normalizing constants dependent on units. We use the following classification loss:

$$L_{\text{Class}} = - \left( \sum_{i \in \psi(\{n\})} \log(c_i) + \sum_{i \notin \psi(\{n\})} \log(1 - c_i) \right) \quad (4)$$

Then our loss becomes:

$$L = L_{\text{MSE}} + L_{\text{Class}} \quad (5)$$

**Sequence training** We also utilize a sequential training method to improve model convergence; rather than immediately training on the entire dataset at once, we pre-train on smaller subsets with varying subtasks. We define three categories of subtasks:

- **Classification:** Predict the number of targets in a given scenario.
- **Fixed target prediction:** Given a fixed number of targets, predict target coordinates
- **Variable target prediction:** Given a variable number of targets (with a known upper bound  $N$ ), predict target coordinates

We begin by training a new model on fixed single-target prediction. Then at each subsequent step, we save the encoder weights and replace the decoder with a new decoder specialized for the next subtask. We gradually increase the number of targets until we reach 1-8 variable target prediction, which is equivalent to the original task. A detailed outline of the training sequence can be seen in Figure 6.

True Positive 95 %	False Negative 5 %
False Positive 11 %	True Negative 89 %

Figure 7. Confusion matrix on simulated dataset

**YOLO** The YOLO model is a state-of-the-art object detection model, providing fast and accurate identification and location of multiple objects in 2-dimensional image. While also based on a CNN architecture, YOLO divides the image into subregions and proposes bounding boxes for potential objects. For our task, we train and test using the YOLOv5 small model.

Our task can be seen as an “edge case” application for YOLO, with only one class, low resolution, and emphasis on coordinate precision. Thus, when converting heatmap data into labelled images, our choice of ground-truth bounding box coordinates are essential. Pixel coordinates for bounding boxes should (1) correspond to “visible” signals on the image, i.e. areas of high signal strength, and (2) continuously and bijectively map to ground-truth azimuth and range coordinates in the original data. Using single-target examples only, we compute a best-fit linear mapping  $f$  from the coordinates  $(x_m, y_m)$  of the center of the highest-magnitude pixel to their corresponding ground-truth values  $(t_m, \phi_m)$ . Then to label the images across the entire data, we use bounding boxes centered at  $f^{-1}(t_m, \phi_m)$  for each true label, with each box four pixels large. By utilizing such a mapping, we ensure that bounded boxes are roughly centered over the corresponding target regions in the image without loss of information between ground-truth coordinates and pixel coordinates.

Training and evaluation details for the general CNN and the YOLO model will be presented in Section 4.

## 4. PERFORMANCE EVALUATION

### 4.1 False Detection

Utilizing an 80/10/10 training/validating/testing split, we use our simulated dataset (4335 examples) to train our classification model for 100 epochs. We additionally provide test results on our real-world dataset (117 examples).

Figure 7 shows the confusion matrix of the simulated validation dataset. The classification model demonstrates 95 % and 89 % accuracy in distinguishing true signals and false signals separately. Figure 8(a) reveals the CLEAN-SAGE could generate up to 3 false detections under a simple LOS-only environment when the algorithm experiences system uncertainty. The confusion matrix of real measurements is shown in Figure 8(b) and achieves 100 % and 98 % accuracy in distinguishing true signals and false signals separately. The validation dataset from real measurements performs better than the simulated test dataset because our measurement is in a controlled LOS environment with all the MPCs being separated in the spatial and time domains. In contrast, the simulated data with multiple MPCs may not be resolved by spatial and time domains representing a more complex environment. The results also validate that the model trained by the simulated could be applied to real measurement if the statistics of system uncertainty are properly modeled.

### 4.2 MPC Extraction

Using an 80/10/10 training/validating/testing split, we train our general CNN model using both our sequence training pipeline and, as a baseline, direct training on the entire dataset. In sequence training, we train on each

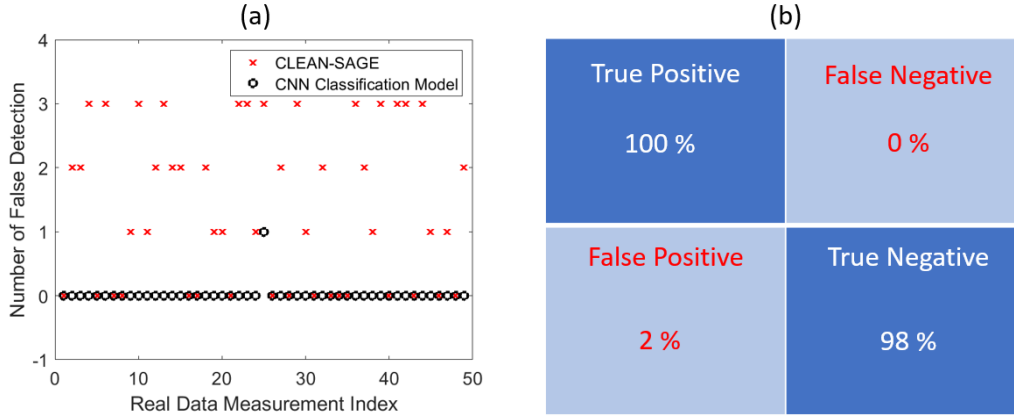


Figure 8. (a) Number of false detections per example in real-world dataset, before and after CNN filtering (b) Confusion matrix on real-world dataset

	CLEAN-SAGE	YOLO	CNN (ladder train)	CNN (direct train)
50 % Error	0.0022(m)	0.0025(m)	0.31(m)	0.89(m)
75 % Error	0.0069(m)	0.0045(m)	0.63(m)	1.9(m)

Table 1. 50 % and 75 % Error

step of the sequence for 3000 epochs, and in direct training, we only train on the full dataset (32000 examples) for 3000 epochs. We train YOLO on the entire dataset for 300 epochs due to runtime limitations.

Figure 9 depicts coordinate precision and average inference time across models. The estimated error in meters has been transferred to log10 first before plotting. YOLO achieves comparable coordinate precision to CLEAN-SAGE, while both general CNN models are less precise by about two orders of magnitude. The 50 % and 75 % errors in meters of each algorithm are shown in the Table 1. To test inference time, we ran all models using a GPU-enabled server. The general CNN model can perform inference on 32,000 examples in 0.508 seconds, while YOLO performs inference on 32,000 examples in 498.756 seconds, and CLEAN-SAGE requires 2663 seconds running on CPU server. In addition, the runtime for both machine learning methods does not depend on target count, making them more scalable than iterative methods such as CLEAN-SAGE. In our experiment, extracting 10 MPC parameters requires 18X more runtime than extracting 1 MPC parameter of CLEAN-SAGE.

Figure 10 depicts coordinate error separated by target count and SNR levels. Under sequential training, the CNN model experiences the highest discrepancy in performance between low-target and high-target cases; this may be due to the compounding nature of the training sequence, where lower-target cases are seen at a higher overall frequency. However, our pretraining sequence still experiences a significant performance improvement over direct training in high-target cases. Notably, the YOLO model is most affected by variations in SNR level; this can be attributed to YOLO’s increased sensitivity to changes in pixel strengths as an image-based approach.

Finally, we note that both YOLO and our CNN can predict the number of targets with high accuracy; we measure this by counting false (prediction which does not correspond to a true target) and missing (true target without a corresponding prediction) targets. When testing over the full dataset, our sequence-trained CNN produces an average of 0.144 % false detections and 0.375 % missing detections rate, while YOLO produces an average of 0.059 % false detections and 2.459 % missing detections rate when using default confidence thresholds. The CLEAN-SAGE with 1.59 % missing detection rate.

## 5. CONCLUSION

In this paper, we developed and investigated deep learning for parameter estimations and false detection for MPC extraction and compared them with the super-resolution method CLEAN-SAGE algorithm. The classification model will be used alongside CLEAN-SAGE successfully reducing the false detections on real and simulated data.

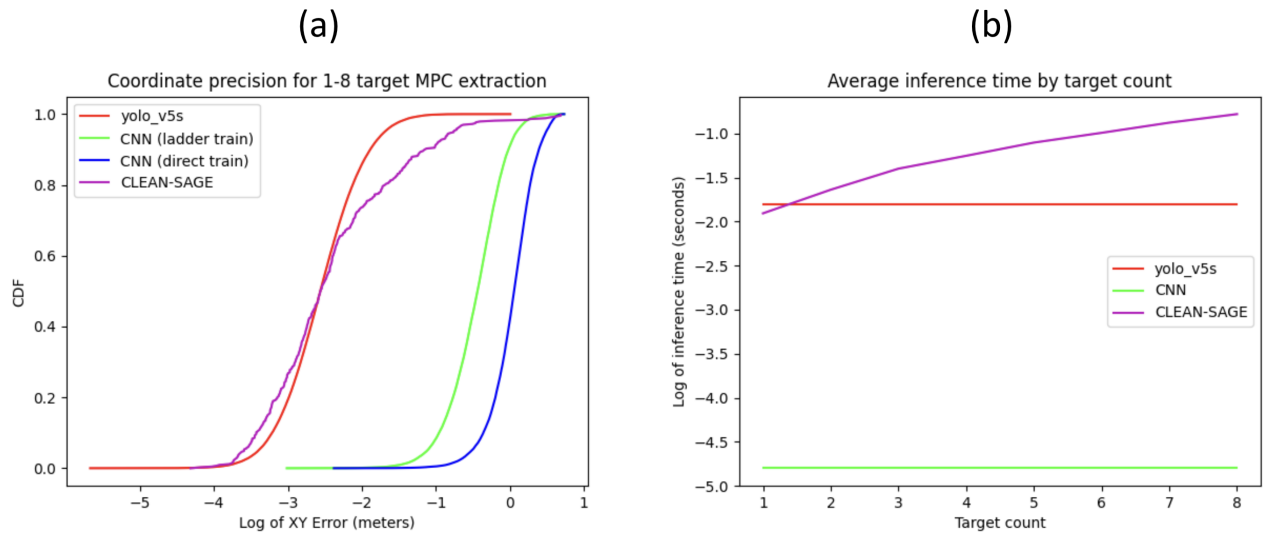


Figure 9. (a) CDF of XY coordinate error per compared target, measured in log (base 10) meters (b) Average runtimes per example, by target count, measured in log seconds

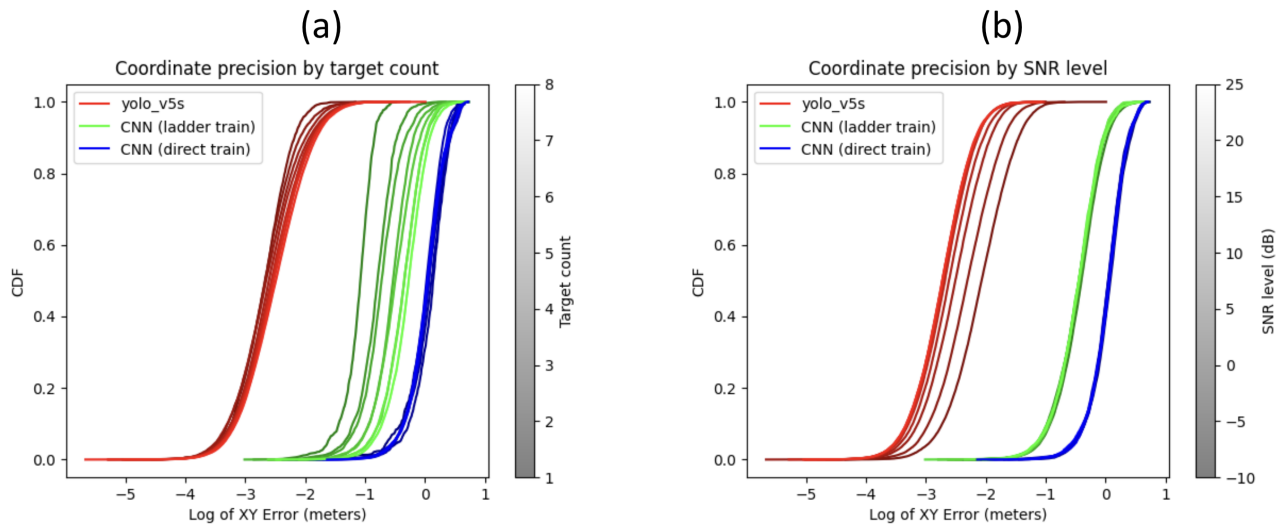


Figure 10. (a) CDF of XY coordinate error per compared target, separated by target count; lighter shades refer to higher target counts (b) CDF of XY coordinate error per compared target, separated by SNR level; lighter shades refer to higher SNR levels

Two regression models have also been developed with a trade-off between parameter estimation error and processing speed. Both regression models demonstrate faster processing time than CLEAN-SAGE while maintaining low false and miss detection rates. The YOLO-based model has comparable performance with CLEAN-SAGE, while the general CNN-based model has a significant speed advantage with a penalty of estimation error. Our initial research reveals the potential and benefit of applying deep learning for parameter estimation for ISAC and channel model applications.

## REFERENCES

- [1] Högbom, J., “Aperture synthesis with a non-regular distribution of interferometer baselines,” *Astronomy and Astrophysics Supplement, Vol. 15, p. 417* **15**, 417 (1974).
- [2] Fessler, J. A. and Hero, A. O., “Space-alternating generalized expectation-maximization algorithm,” *IEEE Transactions on signal processing* **42**(10), 2664–2677 (1994).
- [3] Landmann, M., Kaske, M., and Thoma, R. S., “Impact of incomplete and inaccurate data models on high resolution parameter estimation in multidimensional channel sounding,” *IEEE Transactions on Antennas and Propagation* **60**(2), 557–573 (2011).
- [4] Michelson, D. G., Gentile, C., Molisch, A. F., Chuang, J., Bodi, A., Bhardwaj, A., Ozdemir, O., Khawaja, W. A. G., Guvenc, I., Cheng, Z., et al., “System distortion model for the cross-validation of millimeter-wave channel sounders,” in [2019 13th European Conference on Antennas and Propagation (EuCAP)], 1–5, IEEE (2019).
- [5] Sayeed, A., Guven, D., Doebereiner, M., Semper, S., Gentile, C., Bodi, A., and Cheng, Z., “A framework for developing and evaluating algorithms for estimating multipath propagation parameters from channel sounder measurements,” *IEEE Transactions on Wireless Communications* (2023).
- [6] Huang, C., He, R., Ai, B., Molisch, A. F., Lau, B. K., Haneda, K., Liu, B., Wang, C.-X., Yang, M., Oestges, C., et al., “Artificial intelligence enabled radio propagation for communications—part i: Channel characterization and antenna-channel optimization,” *IEEE Transactions on Antennas and Propagation* **70**(6), 3939–3954 (2022).
- [7] Le Ha, A., Van Chien, T., Nguyen, T. H., Choi, W., et al., “Deep learning-aided 5g channel estimation,” in [2021 15th international conference on ubiquitous information management and communication (IMCOM)], 1–7, IEEE (2021).
- [8] Soltani, M., Pourahmadi, V., Mirzaei, A., and Sheikhzadeh, H., “Deep learning-based channel estimation,” *IEEE Communications Letters* **23**(4), 652–655 (2019).
- [9] Stenin, A. V. and Kalachikov, A. A., “Numerical evaluation of the channel estimation in 5g nr based on machine learning,” in [2022 IEEE 23rd International Conference of Young Professionals in Electron Devices and Materials (EDM)], 285–288, IEEE (2022).
- [10] Delamou, M., Bazzi, A., Chaffi, M., and Amhoud, E. M., “Deep learning-based estimation for multitarget radar detection,” in [2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)], 1–5, IEEE (2023).
- [11] Calist, I. and Kim, M., “Machine learning based channel parameter estimation for indoor environment utilizing reflected rays information,” in [2021 International Symposium on Antennas and Propagation (ISAP)], 1–2, IEEE (2021).
- [12] Wang, J., Chuang, J., Semper, S., and Golmie, N., “Super-resolution localization and tracking in wifi sensing,” in [2024 33rd International Conference on Computer Communications and Networks (ICCCN)], 1–9, IEEE (2024).
- [13] Bang, J., Berweger, S., Chuang, J., Gentil, C., Golmie, N., and Semper, S., “Real-time 141 ghz jacs channel sounder: Near-field switched beamforming, carrier multiplexing, and context aware,” *IEEE Transactions on Microwave Theory and Techniques Submitted Sep-* (2024).
- [14] Luo, C., Wang, J., Feng, G., Xu, S., and Wang, S., “Do deep convolutional neural networks really need to be deep when applied for remote scene classification?,” *Journal of Applied Remote Sensing* **11**(4), 042613–042613 (2017).
- [15] Larson, R. C. and Odoni, A. R., [Urban operations research Ch.7], no. Monograph, Prentice Hall (1981).

- [16] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M., “Learning spatiotemporal features with 3d convolutional networks,” in [*Proceedings of the IEEE international conference on computer vision*], 4489–4497 (2015).
- [17] Zheng, Y., Huang, J., Chen, T., Ou, Y., and Zhou, W., “Cnn classification based on global and local features,” in [*Real-Time Image Processing and Deep Learning 2019*], **10996**, 96–108, SPIE (2019).
- [18] Redmon, J., “You only look once: Unified, real-time object detection,” in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], (2016).