Reducing Training Data for Indoor Positioning through Physics-Informed Neural Networks

Giuseppe Lombardi[a,b], Antonino Crivello[a], Paolo Barsocchi[a], Stefano Chessa[b], Francesco Furfari[a]
[a]Institute of Information Science and Technologies (ISTI), National Research Council (CNR), Pisa, Italy
[b]Department of Computer Science, University of Pisa, Italy

***Abstract*—In this work, we propose a novel framework based on Physics-Informed Neural Networks (PINNs) for directly estimating indoor positions, a method that, to the best of our knowledge, has not been previously explored. Training is performed on a public BLE dataset that includes a variety of indoor scenarios, including Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) conditions caused by human body signal attenuation. The integration of physics-compliant synthetic data during the training phase significantly reduces dependence on large-scale real-world datasets, enabling the use of a simple Multilayer Perceptron (MLP) architecture. Our results demonstrate that combining PINNs with real-world measurements enhances model generalization without compromising accuracy.**

***Index Terms*—PINN, Indoor Positioning, AoA, RSSI**

I. Introduction

Physics-Informed Neural Networks were introduced in 2018 by Raissi et al. as a novel framework to solve forward and inverse problems involving non-linear partial differential equations [1]. PINNs integrate known physical laws, expressed as PDEs, into neural networks by embedding them into the loss function, enabling the solution of complex physical problems even with limited or noisy data. Initially applied to fluid dynamics and heat transfer, PINNs have since evolved to address a wide range of scientific and engineering challenges, including material science, geophysics, and biomedicine [2]. Their ongoing evolution continues to expand their applicability across diverse fields, from energy systems to biomedical engineering.

To the best of our knowledge, there are no direct applications of Physics-Informed Neural Networks for indoor position estimation. In [3], the authors address the problem of optimizing the placement of Wi-Fi access points to improve RSS distribution in complex indoor environments. In [4], a physics-informed approach based on Convolutional Neural Networks (CNNs) is proposed to produce high-resolution power maps aimed at enhancing indoor localization accuracy.

In this work, we employ a Physics-Informed Neural Network to directly estimate indoor position. The network is trained using received signal strength (RSS), azimuthal angle of arrival (AoA), and synthetic data that comply with the differential equations governing the physical model of the system. Training was performed on a public BLE dataset for indoor positioning, which includes measurements collected in various scenarios: one with a constant Line-of-Sight (LoS) between the mobile device and all anchors, another involving human body attenuation, and several cases involving user mobility.

The integration of synthetic data significantly reduces reliance on extensive real-world data for training, enabling the use of a simple Multilayer Perceptron (MLP) network model. By leveraging synthetic data, we demonstrate the feasibility of combining PINNs with real-world indoor localization data, enhancing model generalization while reducing the data collection burden. Furthermore, synthetic data does not compromise performance, which remains excellent while significantly improving the estimates obtained with other localization techniques.

The paper is structured as follows. Section II introduces the physical model adopted for the neural network, its architecture, and the training process. Section III describes the dataset used and the results obtained. The paper concludes with final remarks and future works in Section IV.

II. Modeling the Neural Network Physics

Given a problem with a set of N observations and their corresponding targets, denoted as $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N} \subseteq \mathbb{R}^n \times \mathbb{R}^m$, the objective is to find a function f that maps the inputs $\mathbf{x}_i$ to their respective targets $\mathbf{y}_i \in \mathbb{R}^m$, such that

$$f(\mathbf{x}_i) = \mathbf{y}_i.$$

A neural network with parameters Ω can be trained to approximate this function, yielding a learned function $\hat{f}$ that satisfies

$$\hat{f}(\mathbf{x}; \Omega) \approx f(\mathbf{x}).$$

To achieve this, the optimization process minimizes a loss function with respect to the parameters Ω based on the given observations and their corresponding ground truth values. The choice of the loss function depends on the nature of the task. For classification problems, cross-entropy loss is typically used, whereas for regression tasks, the mean squared error (MSE) is a common choice:

$$L_{\text{data}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{f}(\mathbf{x}_i; \Omega) - \mathbf{y}_i\|^2.$$

The primary objective of the neural network is to generalize effectively to unseen data, even in the absence of ground

This work was founded by European Union - Next Generation EU, in the context of The National Recovery and Resilience Plan, Investment Partenariato Esteso PE8 "Conseguenze e sfide dell'invecchiamento", Project Age-IT, CUP: B83C22004880006

truth labels. However, challenges such as overfitting can arise, particularly when the training dataset is small or when the model complexity is excessively high, potentially leading to poor generalization performance.

To enhance model performance and mitigate overfitting, prior knowledge about the problem can be integrated into the learning process. One effective approach to incorporating domain knowledge is through the introduction of constraints in the form of ordinary differential equations (ODEs) or partial differential equations (PDEs), which govern the physical system underlying the data.

A PDE is a mathematical equation that involves an unknown function f of $n \geq 2$ variables along with some of its partial derivatives. More formally, for an unknown function

$$f : U \subseteq \mathbb{R}^n \to \mathbb{R},$$

where $\mathbf{x}$ lies in an open set U, a k-th order PDE is defined as:

$$F(D^k f, \ldots, Df, f(\mathbf{x}), \mathbf{x}) = 0,$$

where D represents the partial derivative operator.

In scenarios where only limited or no real observations are available, synthetic data points $\{\mathbf{x}_i\}_{i=1}^{C}$, referred to as collocation points, can be generated and utilized in training. These points are passed through the neural network, producing estimates $\hat{f}(\mathbf{x}_i; \Omega)$, where the required derivatives for the physical model are computed automatically using frameworks such as `torch.autograd`. The discrepancy between the neural network's predicted dynamics and the expected physical model is quantified by the residual R, obtained by evaluating the PDE $F(\cdot)$ at the collocation points, incorporating both the model's predictions and their corresponding partial derivatives. This residual provides a measure of how well the neural network adheres to the underlying physical constraints.

Thus, the residual R on the collocation point $\mathbf{x}_i$ is formally defined as:

$$R_i = F(D^k \hat{f}, \ldots, D\hat{f}, \hat{f}(\mathbf{x}_i; \Omega), \mathbf{x}_i).$$

To enforce consistency with the physical model, a physics-based loss function is introduced, defined as the mean squared residual computed over a set of C collocation points $\{\mathbf{x}_i\}_{i=1}^{C}$:

$$L_{\text{phys}} = \frac{1}{C} \sum_{i=1}^{C} R_i^2.$$

The total loss function incorporates both the data loss and the physics loss, ensuring that the neural network adheres to physical constraints while learning from observed data. It is defined as:

$$L = \alpha L_{\text{data}} + \beta L_{\text{phys}},$$

where α and β are non-negative hyperparameters that balance the contributions of data-driven learning and physics-based regularization in the optimization process.

By integrating physics-based constraints, the model not only improves its generalization capability but also ensures compliance with physical laws, leading to a more robust and interpretable learning framework.

A. Path Loss Model and Signal Propagation

A key component of the localization process is the path loss model, which characterizes the attenuation of signal strength as a function of distance. In free space, the received signal strength (RSS) p in dBm at distance r follows the logarithmic path loss model:

$$p(r) = p(r_0) - 10n \log_{10}\left(\frac{r}{r_0}\right),$$

where $p(r_0)$ is the path loss in decibels (dB) at the reference distance r_0, and n is the path loss exponent.

To ensure consistency in the physical model, we impose a governing PDE for the distance r with respect to the received signal strength p:

$$\frac{\partial r}{\partial p} = -k \cdot r, \quad \text{with } k = \frac{\ln(10)}{10n}, \tag{1}$$

expressing that the distance between the receiver and the transmitter increases exponentially as the received power decreases.

B. Position Estimation from RSS and AoA

Given this model, the distance r can be expressed as a function of the received signal strength (RSS) and the azimuth angle of arrival (AoA) θ:

$$r = r(p, \theta).$$

Rather than estimating the distance explicitly, we aim to predict the position of a target transmitter through the unknown function:

$$f(p, \theta) = (x(p, \theta), y(p, \theta)),$$

where (x, y) represents the estimated target position in the two-dimensional plane $\mathbb{R}^2$.

Considering a system with H receiver anchors, each positioned at (x_j, y_j) for $j = 1, \ldots, H$, our goal is to estimate the transmitter's location (x, y) by leveraging the RSS and AoA measurements from each anchor. In other words, we seek to estimate the position given the available information $(p_1, \ldots, p_H, \theta_1, \ldots, \theta_H)$.

Starting from the Euclidean distance formulation:

$$r_j^2 = (x - x_j)^2 + (y - y_j)^2,$$

where r_j denotes the distance between the target transmitter and the j-th anchor, and applying the chain rule, the derivative of r_j with respect to p_j can be expressed as:

$$\frac{\partial r_j}{\partial p_j} = \frac{(x - x_j)}{r_j} \cdot \frac{\partial x}{\partial p_j} + \frac{(y - y_j)}{r_j} \cdot \frac{\partial y}{\partial p_j}.$$

By substituting Equation (1) and rearranging, we obtain a partial differential equation (PDE) involving the spatial derivatives of x and y:

$$(x - x_j)\frac{\partial x}{\partial p_j} + (y - y_j)\frac{\partial y}{\partial p_j} = -k_j \cdot r_j^2. \tag{2}$$

This equation indicates that the scalar product between the vectors $(x - x_j, y - y_j)$ and $\left(\frac{\partial x}{\partial p_j}, \frac{\partial y}{\partial p_j}\right)$ is negative and proportional to the square of the distance.

To integrate the azimuthal AoA information, we use the relation:

$$\theta_j = \arctan\left(\frac{y - y_j}{x - x_j}\right).$$

Applying the chain rule to differentiate θ_j with respect to p_j, we obtain the following PDE:

$$\frac{\partial \theta_j}{\partial p_j} = \frac{(x - x_j)}{r_j^2} \cdot \frac{\partial y}{\partial p_j} - \frac{(y - y_j)}{r_j^2} \cdot \frac{\partial x}{\partial p_j}.$$

Assuming that AoA and RSS measurements are independent for each anchor j, i.e., $\frac{\partial \theta_j}{\partial p_j} = 0$, we obtain the additional PDE:

$$\frac{(x - x_j)}{r_j^2} \cdot \frac{\partial y}{\partial p_j} - \frac{(y - y_j)}{r_j^2} \cdot \frac{\partial x}{\partial p_j} = 0. \tag{3}$$

This equation states that the vectors $(x - x_j, y - y_j)$ and $\left(\frac{\partial x}{\partial p_j}, \frac{\partial y}{\partial p_j}\right)$ are collinear. Consequently, Equations (2) and (3) together imply that $\left(\frac{\partial x}{\partial p_j}, \frac{\partial y}{\partial p_j}\right)$ is proportional to $(x - x_j, y - y_j)$ but oriented in the opposite direction.

We are equivalently stating that signal propagation follows a purely radial motion, with zero tangential component. Therefore, the propagation model satisfies the following system of equations:

$$\begin{cases} F^{(x_j)}\left(\dfrac{\partial x}{\partial p_j}, x\right) = \dfrac{\partial x}{\partial p_j} + k_j(x - x_j) = 0, \\ F^{(y_j)}\left(\dfrac{\partial y}{\partial p_j}, y\right) = \dfrac{\partial y}{\partial p_j} + k_j(y - y_j) = 0. \end{cases} \tag{4}$$

Given a set of collocation points $\{(p_{i1}, \ldots, p_{iH}, \theta_{i1}, \ldots, \theta_{iH})\}_{i=1}^{C}$ and their corresponding position estimates produced by the model:

$$\begin{cases} \hat{x}_i = \hat{x}(p_{i1}, \ldots, p_{iH}, \theta_{i1}, \ldots, \theta_{iH}; \Omega), \\ \hat{y}_i = \hat{y}(p_{i1}, \ldots, p_{iH}, \theta_{i1}, \ldots, \theta_{iH}; \Omega), \end{cases}$$

we compute the residuals of the two differential equations for each anchor and each collocation point as:

$$\begin{cases} R_i^{(x_j)} = F^{(x_j)}\left(\dfrac{\partial \hat{x}_i}{\partial p_{ij}}, \hat{x}_i\right), \\ R_i^{(y_j)} = F^{(y_j)}\left(\dfrac{\partial \hat{y}_i}{\partial p_{ij}}, \hat{y}_i\right). \end{cases}$$

The physics-based loss function is then defined as:

$$L_{\text{phys}} = \frac{1}{2CH} \sum_{i=1}^{C} \sum_{j=1}^{H} \left[\left(R_i^{(x_j)}\right)^2 + \left(R_i^{(y_j)}\right)^2 \right]. \tag{5}$$

This framework integrates RSS and AoA measurements as physical constraints within a physics-informed neural network (PINN) for indoor localization.

Standardizing both input features and target variables improves numerical stability and accelerates convergence during training. Each variable is rescaled to have zero mean and unit variance. Given original variables v and w, their standardized forms are $v^* = \frac{v - \mu_v}{\sigma_v}$, $w^* = \frac{w - \mu_w}{\sigma_w}$. To incorporate this transformation into the governing PDEs, we express derivatives accordingly:

$$\frac{\partial v}{\partial w} = \frac{\partial(\sigma_v v^* + \mu_v)}{\partial(\sigma_w w^* + \mu_w)} = \frac{\sigma_v}{\sigma_w} \cdot \frac{\partial v^*}{\partial w^*}.$$

This change ensures that the model operates in a normalized domain, enhancing efficiency and accuracy in PINN-based training [5].

C. Neural Network Architecture

The proposed model is based on a multi-layer perceptron (MLP) architecture. Figure 1 provides a graphical overview of the architecture, illustrating the flow of input features through the hidden layers to the final output coordinates. The network is designed to be fully compatible with the physics-informed framework described in the previous sections.

The input to the model is a $3H$-dimensional feature vector designed to capture key signal characteristics, where H is the number of anchors. The first H elements represent the received signal strength (RSS) from the anchors. The next H elements store the cosine values of the azimuthal angles of arrival (AoA), while the final H elements contain the corresponding sine values. Representing angles through their cosine and sine components, rather than using raw angular values, ensures a continuous and smooth representation. This approach prevents issues associated with angular discontinuities, such as the transition between 0 and 2π, and instead provides a robust representation of directional information as a unit vector in the plane. The network has two hidden layers, each with $2H$ neurons, where H is the number of anchors. This configuration is designed to extract and integrate information from the RSS and AoA of each anchor. To capture the non-linear component inherent in the localization task, the hidden layers use the hyperbolic tangent (`tanh`) activation function. An output layer then maps the learned internal representation to the estimated position (x, y) in the plane.

D. Training Process

The model is trained using the AdamW optimizer, which integrates momentum-based updates with weight decay regularization. Unlike the conventional Adam optimizer, AdamW decouples weight decay from the loss function, directly penalizing the L^2-norm of the network weights without modifying the loss computation. A learning rate of 0.1 is used, with a weight decay parameter of 0.01 to regulate weight magnitudes, improving convergence and mitigating overfitting.

The observed data used to train the model are partitioned into training and validation subsets using an 80/20 split. To prevent overfitting, an early stopping strategy with a patience of 10 epochs is implemented, ensuring that the model training halts if the validation loss does not improve for 10 consecutive epochs. The model chosen for deployment corresponds to the epoch with the lowest validation loss. Additionally, a learning

rate scheduler is employed to enhance stability, reducing the learning rate by a factor of 0.1 if the validation loss stagnates for five consecutive epochs. Standardization is applied to both input features and ground truth coordinates using the mean and standard deviation computed from the training dataset. This preprocessing step accelerates model convergence, and after inference, the predictions are de-standardized to their original scale before evaluation.

During training, the observed data are processed in batches of size 256. For each batch, the model is trained using real-world data from the batch and synthetically generated collocation points. The generated collocation points are used to compute the residuals of the governing differential equations, ensuring that the model adheres to the underlying physical constraints. Therefore, the total loss is computed as the sum of two components: the data loss from real measurements and the physics-based loss (Equation 5) derived from synthetic data. The model then calculates the gradient of this combined loss function and updates its parameters accordingly.

Since the received signal strengths (RSS) and angles of arrival (AoA) are independent for each anchor but inherently dependent on the position (x, y) in the indoor environment, the synthetic data are generated using a structured approach. A grid of $n = 1024$ points is defined over the indoor space, and for each point, the RSS values from the anchors are computed based on a path loss model, incorporating the 1-meter RSS reference values obtained from calibration. The azimuthal arrival angles are determined mathematically from the known anchor positions. To simulate real-world measurement uncertainty and mitigate overfitting to the physics-based constraints on synthetic data, Gaussian noise is added to the RSS and AoA values and resampled every $k = 8$ epochs to regenerate the synthetic collocation points. The added Gaussian noise has zero mean, with standard deviations of $\sigma_p = 2\,\mathrm{dBm}$ for the signal powers and $\sigma_\theta = 5^\circ$ for the angles of arrival.

III. Experimental Results

Experiments were carried out using a dataset from a prior study [6], accessible publicly [1]. The tests were performed in a large indoor space free of obstacles. Four anchor nodes were positioned midway along each wall, mounted on tripods at a height of 2.3 meters. The area enclosed by these nodes measured 6×12 meters. Azimuth and elevation measurements were recorded in even degrees, indicating a precision of 2 degrees. This results in a positional discrimination limit of $35cm$ at a distance of 10 meters.

The dataset encompassed four scenarios: calibration, static, mobility, and proximity. This study focused on calibration, static, and mobility scenarios. In the calibration scenario, data were gathered using a 120 cm-high tag placed at 119 locations within the test area, uniformly distributed and always in line of sight (LoS). The static scenario included four datasets based on the orientation of the participant, with data collected from a tag worn by an individual at 36 positions. These positions were a subset of the 119 used in the calibration scenario. The tag was worn around the neck (1.12 m from the ground), and data were collected with the individual facing North, South, East, and West to evaluate the effect of body orientation on the measurements. For each orientation, only one anchor remained in LoS.

A. Data preprocessing

Before training, the data collected by each anchor undergoes preprocessing to ensure consistency across measurements. Since the anchors operate asynchronously, their signals must be merged into a unified dataset. To achieve this, signals from the four anchors that fall within a predefined time buffer

[1] https://zenodo.org/record/7759557

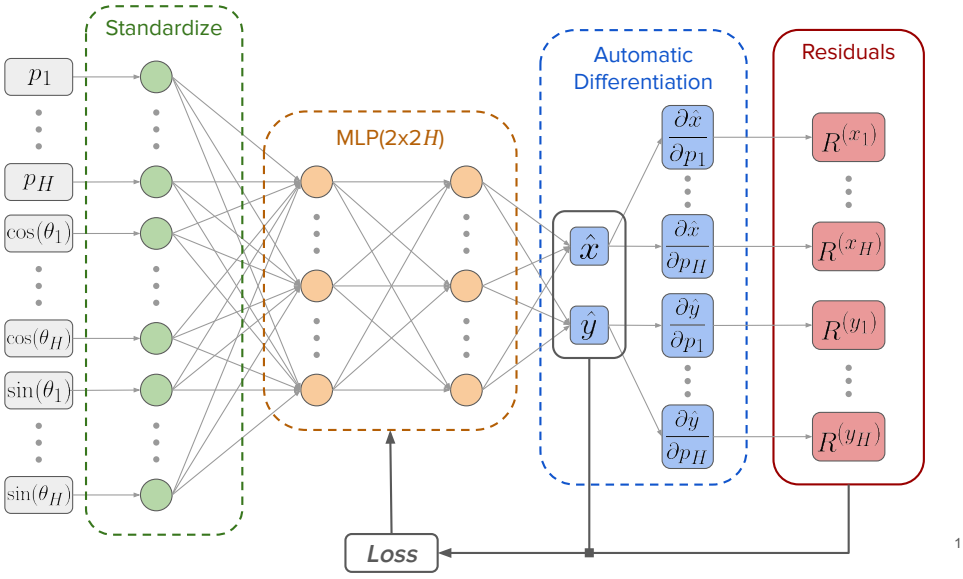

Fig. 1. Graphical representation of our PINN model for Indoor Positioning.

are aggregated, retaining only the most recent measurement for each anchor within the buffer. For calibration and static datasets, a time buffer of 1000 ms is applied to ensure stable positioning, while for mobility datasets, a shorter time buffer of 200 ms is used to capture dynamic movements more accurately. Additionally, in mobility datasets, only signals within the time window $[t_i - 400, t_i]$ are considered, where t_i represents the timestamp corresponding to the subject's passage through the ground truth position (x_i, y_i). The 400 ms window approximates the duration of a single step, allowing for precise evaluation of the model's performance in motion scenarios.

Following the preprocessing phase, the number of resulting observations is approximately 7300 for the calibration scenario and 2350 for each static scenario. Among the original datasets, we include *Static All*, which aggregates data from all static scenarios in which the subject is oriented toward each of the four cardinal directions, resulting in approximately 9400 observations.

B. Results and analysis

This section presents the main results using RMSE to compare PINN performance across different datasets. At the end, CDFs are provided to assess overall localization performance.

Table I presents two key results. The first concerns the impact of altering the ratio between training and testing data, and the second concerns the use of synthetic data with a PINN.

The first result is illustrated by the second column (highlighted in red). It shows the degradation in estimation accuracy when the training set is reduced to 20% of the original data while the remaining 80% is used for testing. In this case, the overall accuracy decreases by 18%.

TABLE I
THE TABLE PRESENTS THE RMSE RESULTS (IN CM). COLUMNS INDICATE THE SPLIT RATIO OF DATASETS IN TRAIN AND TEST.

	Data only		**Data + Physics**
Scenario	80% Train 20% Test	20% Train 80% Test	20% Train 80% Test
Calibration	30	37	36
Static East	49	58	52
Static North	52	55	54
Static West	42	50	47
Static South	55	72	64
Static All	79	90	81

The second result is illustrated in the third column (highlighted in blue). It reports the outcomes obtained when training is still performed on only 20% of the original data, but augmented with synthetic data. In this case, we exploit the PINN to recover accuracy, achieving an average improvement of 8% in NLoS scenarios. For the Calibration dataset, however, improvements are negligible, suggesting that the proposed solution is more effective in real-world scenarios where users actually wear the devices.

Table II highlights two additional key aspects. The first pertains to the network's performance when trained on 100% of the data from the Calibration dataset and tested on different datasets characterized by NLoS conditions. In this setup, the network is trained on highly stable data, as the tag was always in line of sight during the data collection. However, testing is conducted on more challenging data, where signal perturbations arise due to attenuation caused by the presence of the human body.

TABLE II
THE TABLE PRESENTS THE RMSE RESULTS (IN CM), BY TRAINING AND TESTING ON DIFFERENT DATASETS.

	Data only	**Data + Physics**
Test Scenario	Trained on Calibration	Trained on Calibration
Static East	177	128
Static North	139	127
Static West	198	119
Static South	163	161
Static All	170	134

As expected, under these conditions, the network does not achieve the same level of performance observed in previous experiments. Nevertheless, when the Calibration dataset is augmented with synthetic data and used to train the PINN, performance improves significantly. The second column of the table reports the PINN error (RMSE) for each scenario, showing an average improvement of 21%.

The experiments reported in Table III was conducted on the mobility dataset. While the test should ideally use differential equations that capture motion, it is still useful to assess how the network behaves when trained on static data. Initially, we trained the MLP network using 100% of the data from both the Calibration and Static All datasets. Training on Static All proved to be more effective than training on Calibration, as it includes perturbations collected with the person oriented in all cardinal directions, which better reflect the variability encountered in mobility paths. Improvements were observed for both datasets, although training with Static All yielded more accurate estimates.

TABLE III
THE TABLE PRESENTS THE RMSE RESULTS (IN CM). ROWS REPRESENT THE TESTED MOBILITY USE CASES.

	100% Data only		**20% Data + Physics**	
Test	Trained on Calibration	Trained on Static All	Trained on Calibration	Trained on Static All
Use case 1	129	87	94	80
Use case 2	152	65	95	66
Use case 3	200	123	156	153

Figure 2 shows the cumulative distribution function (CDF) of the localization error, comparing the performance of the model trained solely on observed data with that of the physics-informed model. Three evaluation tasks are considered in this comparison. The first task, [C20:C80] (solid line), uses 20% of the *Calibration* data for training and 80% for testing. The second, [S20:S80] (dotted line), uses a 20/80 split of the *Static*

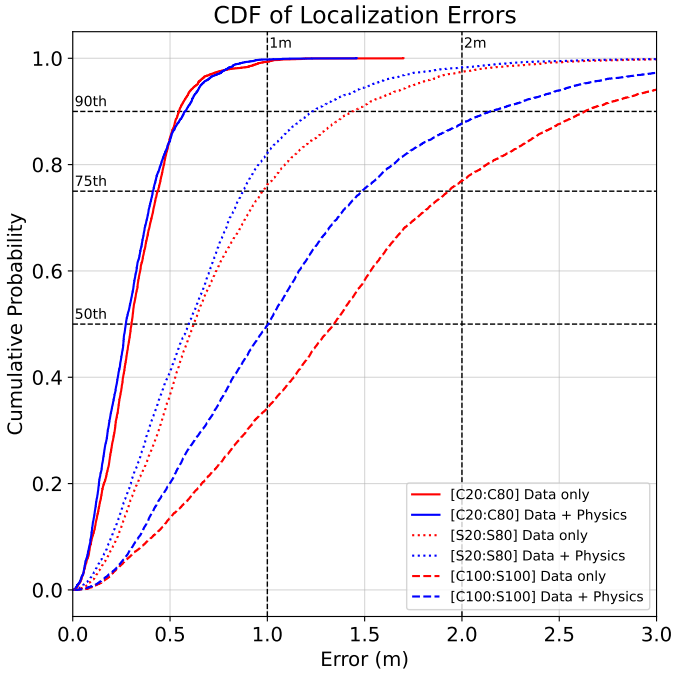

Fig. 2. CDF of localization error comparing the data-only model and the physics-informed model across multiple tasks.

All scenario. The third, [C100:S100] (dashed line), trains on all *Calibration* data and tests on all *Static All* data.

An initial analysis of the results reveals that in the Calibration scenario – where there is no perturbation from the human body – the model trained only on observed data is less affected by the reduced training set size. In this case, the error percentiles of the two models are very similar, with the 50th, 75th, and 90th percentiles at approximately 28 cm, 42 cm, and 56 cm, respectively.

The impact of the physics-informed component becomes more significant in the Static All scenario when the training set is reduced. Due to increased signal variability introduced by the human body, the expressiveness of the training data is limited after the reduction of the observations. Here, the physics-informed model outperforms the data-only model, with localization errors of 60 cm at the 50th percentile, 88 cm at the 75th, and 123 cm at the 90th percentile. This corresponds to improvements of 3 cm, 10 cm, and 21 cm, respectively, over the data-only baseline.

The greatest benefit of incorporating physical knowledge is observed in the cross-scenario task [C100:S100]. A model trained solely on Calibration data performs poorly when tested on Static All due to the distribution shift caused by the presence of the human body. In this setting, the physics-informed model shows a markedly improved CDF curve, with localization errors of 100 cm, 149 cm, and 214 cm at the 50th, 75th, and 90th percentiles, corresponding to reductions of 34 cm, 44 cm, and 49 cm, respectively, compared to the data-only model.

IV. Conclusion and Future Works

Classic machine learning models rely entirely on data, requiring a sufficient quantity of well-distributed, low-noise data to effectively learn a given task. In the context of indoor localization, acquiring a dataset with accurate ground truth labels is essential yet time-consuming, as it must sufficiently cover diverse scenarios to ensure reliable model performance.

By incorporating physical knowledge of signal propagation and geometric properties of indoor spaces, it is possible to reduce the model's reliance on extensive training data. Physics-informed neural networks (PINNs) help mitigate performance degradation in cases of limited or noisy data by introducing a natural regularization effect during the learning process. The use of carefully generated synthetic data allows the network to respect the physical laws governing signal propagation in the transmitter-receiver model, avoiding the introduction of unintended biases that may arise from training solely on artificially labeled data.

Real-world measurements, in this context, serve as initial conditions for the differential equations defining the physical model, enabling the adaptation of the model to observed data.

The application of this approach to planar indoor positioning naturally extends to three-dimensional localization, opening avenues for estimating the z-coordinate in indoor environments. Additionally, the generation of synthetic data consistent with the underlying physics remains an ambitious goal, with potential for further refinement through various strategies. Another promising direction is increasing the complexity of the physical model by accounting for intricate indoor spaces, obstacles, and human body interference. Finally, an extension of this work involves introducing temporal dependencies into the model by incorporating kinematic constraints, thereby improving position estimation in dynamic scenarios where mobility plays a critical role.

References

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2017.

[2] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[3] D. Cui, G. Yang, S. Ji, S. Luo, A. Seretis, and C. D. Sarris, "Physics-informed machine learning models for indoor wi-fi access point placement," in *2021 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting (APS/URSI)*, 2021, pp. 227–228.

[4] F. Ashqar, R. Khoury, C. Wood, Y.-H. Yeh, A. Seretis, and C. D. Sarris, "Physics-informed convolutional neural network for indoor localization," in *2021 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (APS/URSI)*, 2021, pp. 659–660.

[5] S. Xu, Y. Dai, C. Yan, Z. Sun, R. Huang, D. Guo, and G. Yang, "On the preprocessing of physics-informed neural networks: How to better utilize data in fluid mechanics," *Journal of Computational Physics*, vol. 528, p. 113837, May 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999125001202

[6] M. Girolami, F. Furfari, P. Barsocchi, and F. Mavilia, "A bluetooth 5.1 dataset based on angle of arrival and rss for indoor localization," *IEEE Access*, vol. 11, pp. 81 763–81 776, 2023.