2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) | 979-8-3503-6224-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/PIMRC59610.2024.10817452

Deep Hypernetwork-based Robust Localization in Millimeter-Wave Networks

Roman Klus*, Jukka Talvitie*, Benjamin Domae†, Danijela Cabric†, and Mikko Valkama*
*Department of Electrical Engineering, Tampere University, Tampere, Finland
†Electrical and Computer Engineering Department, University of California, Los Angeles, USA

***Abstract*—Wireless localization and sensing are increasingly important capabilities when the networks are evolving towards the 6^{th} generation era. While the physics-inspired geometrical models are known to perform well in line-of-sight (LoS) dominant scenarios, harnessing the power of artificial intelligence (AI) to improve robustness, efficiency, and performance in more complex propagation scenarios is an intriguing prospect. To this end, the hypernetwork (HN) is an emerging neural network (NN) architecture, where one model is used to parameterize the weights of the other, promising dynamic weight adaptation among other performance improvements. In this work, we propose the concept of Hypernetwork Localization (HypLoc) – a hybrid HN-based architecture for localization in beamforming millimeter-wave (mmWave) networks, while combining angle-of-arrival (AoA), time-of-flight (ToF), and received power (RP) as representative measurements. Considering a realistic urban vehicular environment, we first demonstrate the baseline effectiveness of HypLoc with a fixed and known gNodeB (gNB) deployment scenario. We then also study a scenario where the factory pre-training covers multiple different gNB deployment constellations and show that the proposed HypLoc clearly outperforms the traditional NNs. Finally, we also show that the HypLoc adapts faster and requires less training data when adapting to a previously unseen deployment scenario. Overall, the proposed approach facilitates efficient factory pre-training when operating under multiple different gNB deployment options.**

***Index Terms*—Deep learning, factory pre-training, hypernetwork, localization, mmWave networks, vehicular systems**

I. Introduction

One of the key enablers and assets of the next-generation wireless networks will be integrating artificial intelligence (AI) technology into various network functionalities [1]. To this end, pursuing machine learning (ML) and neural network (NN) solutions within the different protocol layers and network capabilities, outperforming classical model-based solutions is one timely and active research challenge [2]–[7]. In this regard, high-accuracy localization and sensing represent one important application and capability domain where ML and NN based approaches are under active research [8]. The emerging location-based services (LBS) that require more and more accurate and reliable localization [9], together with the requirements for seamless mobility management and handover procedures, catalyze the technology development and research innovations in this field.

Utilizing data-driven models in wireless localization over traditional model-based ones is of high interest within the research community, and numerous works introducing off-the-shelf NN models within the fifth generation (5G) and beyond networks have been published. To this end, a user equipment (UE) positioning system utilizing the beam-specific reference signal received power (RSRP) values as the features was presented in [10]. Signal propagation time as a feature has also raised interest in numerous works including [11], where time-of-arrival (ToA) measurements were used with a recurrent NN to track mobile users such as connected vehicles. Moreover, beamforming active antenna arrays in millimeter-wave (mmWave) networks allow for directional signal transmission and reception, enabling angle-based cellular positioning within NN-driven systems [12]. Furthermore, combining numerous network measurements have been shown to be beneficial when utilizing NN models in complex non-line-of-sight scenarios [7].

Factory pre-training is one of the promising concepts for enabling the deployment of the AI-driven base stations (BSs), such as 5G gNodeBs (gNBs), without lengthy on-site data acquisition and subsequent NN model training. Generating representative synthetic data is cheaper and faster, while models trained on such data are capable of reaching competitive performance [13]. For example, [14] introduced a paradigm to generate a high-accuracy 5G localization dataset, considering a realistic ray-tracing model and channel frequency response as the actual feature. Apart from generating representative and reliable data, the capabilities of the pre-trained models are determined by the selected NN model architecture. The utilization of hypernetworks (HNs) is an emerging concept carrying strong potential while still being only marginally explored, especially in the communications domain. The HN approach considers a pair of NNs, one generating the parameters for the other, and is expected to enable dynamic model adaptation and improved performance, while also allowing combination with any type of NN structure [15].

A survey on HNs was presented in [15], introducing basic concepts and applications, while [16] provides several hands-on experiments, combining HNs with convolutional and sequential models. A multiple-input multiple-output (MIMO) detector implemented as a sequential HN model was proposed in [3] with promising results. Similarly, to avoid the necessity to re-train the whole NN-based detector for each channel realization, [17] proposes a HN structure to dynamically adapt the model weights. HN was also utilized for a non-stationary channel prediction in [4], adapting the model based on the received

This work was supported by the Research Council of Finland under the grants #352754, #357730, and #359095. This material is also based upon work supported by the National Science Foundation under Grant No. 2224322.

979-8-3503-6224-4/24/$31.00 ©2024 IEEE

channel state information (CSI). A joint source-channel coding using hypernetworks was introduced in [5], taking channel conditions as inputs and generating weights for the channel encoder and decoder at the same time, improving model adaptability, memory efficiency, and transmission accuracy. HN was used to improve the channel estimation in a mmWave system in [6]. Utilizing a HN to estimate the parameters for different models than NNs was attempted in [18] and [19], combining the concept with expectation consistent signal recovery algorithm and a Kalman filter, respectively.

While the HN is an emerging topic in the field of wireless communications, this is the first localization-oriented study utilizing and exploring this promising technique. The proposed localization model consists of a HN structure combined with a traditional NN model to enable seamless, efficient, and accurate positioning performance. Specific emphasis in the study is on improved robustness against different gNB deployment constellations while also seeking to generalize towards unseen deployments. One important application scenario of the proposed method is localizing and tracking vehicles in complex built environments with complicated propagation characteristics that hinder the efficiency of classical methods.

The main technical contributions and novelty of the work can be summarized as follows:

- We introduce and discuss the main concepts of the HN structure and propose the Hypernetwork Localization (HypLoc) concept, a novel HN-based architecture for wireless localization;
- We evaluate the feasibility of utilizing HypLoc as a positioning model in a well-established urban environment, incorporating cars and other vehicles as UE, and adopt accurate ray tracing tools for realistic propagation modeling in the considered 28 GHz mmWave network;
- We study the advantages and drawbacks of the HN architecture within HypLoc as compared to traditional NN models in terms of positioning performance, generalization capabilities, and adaptability to new data.

The rest of this paper is organized as follows: Section II discusses the basic concepts of HNs and presents the considered 5G network positioning measurements. Section III introduces the technical specifics of the proposed model, in terms of architecture, hyperparameters and training, while Section IV provides the numerical evaluation results, along with the details on the evaluation scenario and evaluated models. Finally, Section V concludes the work.

II. Hypernetworks and 5G Positioning

Apart from describing the model determining the parameters of the main network, the term hypernetwork is interchangeably used within the literature to describe the pair of hypernetwork and the main model as a structure. For terminological clarity within this manuscript, the term hypernetwork (HN) denotes the general concept and the model, while the pair of HN and the main model is referred to as hypernetwork structure (HNS).

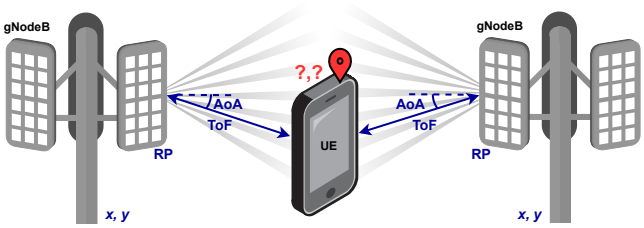

Fig. 1. General concept of wireless positioning, where UE is localized based on measured channel or received signal features at multiple gNBs.

A. Hypernetworks

While the basic concepts of deep learning are widely known [20], the concept of HNs has only recently emerged within the scientific domain. The basic idea of a HNS consists of two NNs, where one, called a HN, determines the weights and biases (parameters) for the second, called main or primary network. Both networks can be trained in an end-to-end manner [15]. Utilizing a HNS enables several crucial things, which include reducing the number of trainable parameters (thus reducing the training complexity) [16], enabling near-optimal weight initialization [15], improving the overall performance [6], and adjusting the model's weights in each iteration of the model [17], effectively adapting the NN to the given circumstance or environment.

Formally, while the standard NN model can be interpreted as a function $F(X, \Phi)$, which is trained through backpropagation while minimizing the loss function $L(l, \hat{l})$ where $l = (x, y)$ denotes labels and $\hat{l} = (\hat{x}, \hat{y})$ label estimates, the HNS can be similarly expressed as $F(X, H(Z, \Psi))$, where $H(Z, \Psi))$ denotes the HN model. Here, X denotes the input features of the NN, Z denotes the input features of the HN, and Φ and Ψ denote the trainable parameters composed of weights and biases of each network, respectively. The difference between the two models has clear and direct consequences regarding their behavior. By exchanging Φ for $H(Z, \Psi))$, the structure transforms the static weights to a set of variables, which may change based on the input Z. Thus, HN can adaptively change the behavior of the main model, leading to greater generalization properties, as well as implementing the degrees of freedom within the model itself.

B. Positioning Concept and Considered 5G Measurements

Utilizing a NN for radio-based positioning is realized by feeding the network measurements to a positioning model, which outputs the location estimates of the user. Generally, any type of measurements can be utilized, ranging from received power [10], angular information [12], propagation time [11], to channel frequency responses [21] and their combination [7], as depicted in Fig. 1. Multiple gNBs receive the signal transmitted by the target UE and variations in their measurements due to different signal propagation enable the model to learn how to localize the user. To learn the NN model, a so-called training database needs to be acquired with representative and

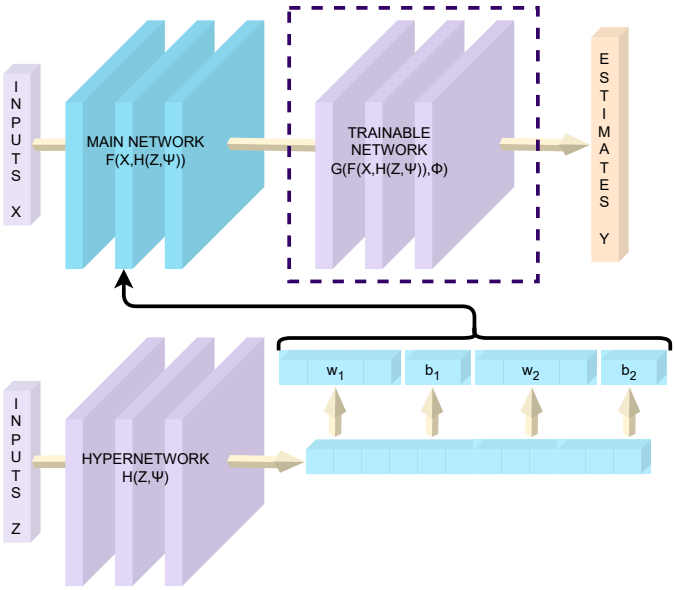

Fig. 2. Design of the overall HNS, where the HN estimates the parameters of the main network. The addition of a trainable network within the proposed HypLoc structure transforms the HN and main network models into powerful feature extractors.

location-tagged measurements, which can be either measured or synthetically acquired via simulations.

The considered physical quantities related to the received signal and the underlying radio channel are angle-of-arrival (AoA), time-of-flight (ToF), and received power (RP). In practical networks, gNBs equipped with antenna arrays are capable of harnessing information on distinct radio propagation paths in terms the ToF and AoA (at the gNB) through multi-round trip time (MRTT) measurements utilizing uplink (UL) sounding reference signal (SRS) and downlink (DL) positioning reference signal (PRS) [22], [23]. In the 5G standard, the considered measurements can be obtained by following the procedures and specific signalling structures related to the MRTT and uplink angle-of-arrival (UL-AoA) positioning [23].

III. Proposed Methods

A. Hybrid Model Architecture

In this work, we seek to especially exploit the generalization properties of the HNS to adapt the positioning model to the variations in the deployment coverage and extend the generalization capabilities to the previously unseen scenario. We include an additional model, denoted as the trainable model, in a series after the main network, as shown in Fig. 2, to alleviate some of the HN's drawbacks discussed later in the text. The general HNS is specified in Fig. 2 when not considering the separated trainable model.

In the proposed HypLoc architecture only a part of the main data pipeline is parametrized by the HN, while the other part remains static. With this solution, we aim to create a robust model incorporating the advantages of both the static model (trainable network) and the HNS (main network + HN pair). To this end, following the general structure depicted in Fig. 2, the data flow within the model is as follows: the inputs Z are passed through the HN, which estimates the weights and biases for the main network. The main network is then parametrized and the inputs X are then passed through it. The outputs of the main network are fed to the static, trainable network. Following the notation in Section II-A, the trainable network considers the main network's outputs $F(X, H(Z, \Psi))$ as input features, while its parameters Ψ remain static. In the training process, the loss $L(\cdot)$ is calculated between the estimates and the labels, and by applying the backpropagation through the whole system the trainable parameters Ψ and Φ are updated in both the trainable network and the HN, since all operations within the structure are differentiable.

B. Related Challenges

Utilizing a HN brings several challenges that need to be either considered or directly targeted. Among the most significant ones is the large number of trainable parameters and weights the HN has to generate, despite the relatively small size of the main network. More concretely, each neuron in a NN contains $D_in + 1$ parameters, where D_in denotes the dimension of its input and 1 refers to that neuron's bias. Consequently, a simple, single-layer main network with 100 inputs and 50 neurons requires $50 \cdot (100+1) = 5050$ parameters to be provided by the HN. In turn, the HN requires as many neurons in its last layer, i.e. 5050, while the previous HN's layer needs to be adequately large to pass enough information, e.g. consisting of 500 neurons (which is understated to say at least). Consequently, only the last layer of the HN consists of $5050 \cdot (500 + 1) = 2530050$ trainable parameters to operate a single, 100-neuron main network. Maintaining the dimensions of both networks is thus a critical objective to consider and overcome.

Carefully optimizing the training algorithm of the HNS is of critical importance. While the weights of the traditional NNs are being iteratively adjusted in very small instances, the HN needs to ensure that the weights it provides are consistent, stable, and efficient. Carefully selecting the hyperparameters of the model and tracking the training progress is crucial, as HN is susceptible to inaccuracies and biases. Furthermore, ensuring the stability of the training data further enables seamless operation of the HN. Uncertainties, inconsistencies, and outliers, all can lead to performance degradation.

C. HypLoc Hyperparameters and Training

To keep the main and trainable network powerful yet lightweight while maintaining the HN parametrization in a reasonable range, the main network was constructed as a single-layer, densely connected model with 32 neurons and a Gaussian error linear unit (GELU) activation function [24]. The same input vector is considered for both the main network and the hypernetwork, i.e., $X = Z$, and consists of 56 elements in total (described in Sec. IV-A). Consequently, the HN has to provide 1824 parameters to the main network. To address the challenge of training parameters' count, the HN was experimentally established as a 3-layer NN with 64, 128, and

TABLE I: HYPLOC ARCHITECTURE - BASIC INFORMATION

Model	Layers	Neurons	Parameters	Trainable parameters
Main model	1	32	1824	0
Hypernetwork	3	2016	247264	247264
Trainable model	3	66	2178	2178
Total	7	2114	251266	249442

1824 neurons, which limits the number of trainable parameters to 247264, out of which 235296 are within the last layer. The first two layers have GELU activation, while the third considers a sigmoid activation to limit and stabilize the output. The trainable network is a 3-layer, densely connected NN with 32, 32, and 2 neurons, respectively. The first two layers consider GELU activation, while the output layer has no activation to enable a continuous range of outputs.

The overall information and hyperparametrization of each NN are summarized in Table I, which clearly shows that the HN is by far the most complex model within HypLoc.

The HypLoc model was trained for 30 epochs in total, with Adam optimizer specified with learning rates starting at 0.001 and decreasing by 4% for every 10000 optimizer steps (processed samples). Mean squared error was selected as a quadratic loss function, consistent with the positioning error. The decaying learning rate first allows all models to adapt to the environment and the data and later enables the hypernetwork to stabilize its outputs.

IV. NUMERICAL RESULTS

A. Evaluation Scenario and Assumptions

The evaluation environment utilized in this work is realized using ray-tracing-based channel measurements based on Wireless Insite®software [25]. We employ the map-based Madrid grid proposed by the METIS society [26], recognized as the relevant urban scenario by 3rd Generation Partnership Project (3GPP) in 5G New Radio (NR) specifications [27]. The Madrid grid layout introduces generally a rich radio propagation environment with varying street widths and open areas, which provide solid generalization and scalability options for the model trained in the considered environment and setup.

The simulated urban scenario contains eighteen 5G mmWave gNBs, illustrated along Fig. 3 with red rectangles. Each gNB is equipped with a uniform cylindrical antenna array placed at 5 m height. The network is configured to operate at 28 GHz carrier frequency. The AoA and ToF measurements are obtained based on the corresponding characteristics of the radio propagation path with the highest received power, building on the signals and measurement procedures described in Section II-B. The obtained AoA measurements are acquired with a 20 degrees resolution, reflecting the limited directional resolution of the gNB. The network measurements corresponding to the samples with path-loss higher than 160 dB are not considered, while in general, the environment shown in Fig. 3 possesses large areas and street segments with severe multi-bounce phenomena.

The aggregate dataset comprises 36 vehicle-like user trajectories, where the mobile UE captures data at 100 ms intervals.

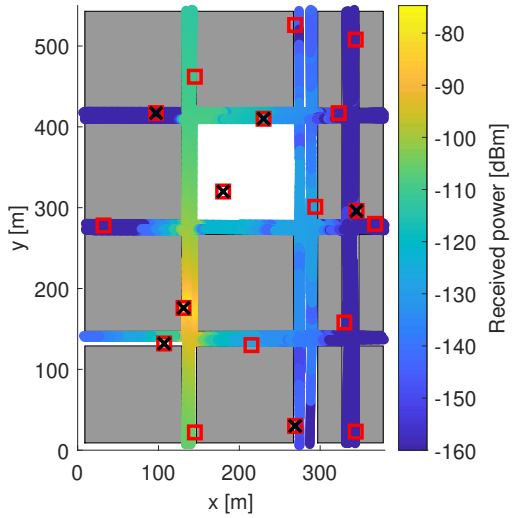

Fig. 3. Considered Madrid map ray-tracing environment with a total of 18 gNBs marked with red rectangles, out of which 7 are active in a given constellation. One example constellation is illustrated with black crosses. The figure also shows the distribution of the received power of a single gNB.

For each trajectory, the initial UE position is set randomly along the streets. UEs navigate within the region with an equal probability of moving in any direction at intersections. The UE speed ranges from 20 km/h to 60 km/h, contingent on the current street and the proximity to intersections. When nearing an intersection, the UE decelerates at a rate of 3 m/s^2 until it reaches a constant speed of 20 km/h to ensure smooth turning. Post-turn, the UE accelerates at 2 m/s^2 until it hits the street-specific speed limit. The standard speed limit is set at 40 km/h, except for the right-most vertical street, which has a limit of 20 km/h (refer to Fig. 3), and the broader street next to the pedestrian area, which is capped at 60 km/h. While all user trajectories adhere to right-hand traffic rules, the specific UE paths, and corresponding measurement points differ across the simulated trajectories.

Furthermore, this work considers numerous different gNB deployment constellations, each consisting of 7 active base stations selected randomly out of the available 18. For example, constellation #1 is denoted in Fig. 3 with black crosses. Out of 36 available user tracks, 4 were selected for testing, while the remaining 32 were combined to create a training set. Each sample is considered individually, with 22775 samples in the training set and 2406 samples in the test set. Each sample is represented by received power, ToF, and AoA in the form of three vectors representing its x, y, and z components for each gNB, with an additional x, y, and z coordinates of the corresponding gNB. In total, each sample is a vector of $7 \cdot 8 = 56$ elements.

B. Benchmark Models

To fairly assess the performance and effectiveness of the proposed structure, we utilize several reference NN models, each providing valuable insights into the HN's behavior. The first considered architecture is a NN model constructed as a feedforward NN with 3 intermediate layers of 32 neurons each,

and a 2-neuron output layer. By following the architecture of a main network and the trainable networks, the model's performance corresponds to utilizing only the feedforward pipeline of the proposed structure and consists of 4002 trainable parameters. The second considered architecture is constructed as a feedforward NN with 3 intermediate layers of 340 neurons each, and a 2-neuron output layer. This model, in turn, represents the classical model with similar complexity to the proposed HN structure, as it has 251942 trainable parameters. Moreover, the parameters of all reference NN models have been briefly optimized.

In the evaluation, we often utilize multiple gNB constellations and we either train a single model with all available data or create multiple models, each corresponding to an individual deployment constellation. The term *single model* denotes that one instance of the model was trained with all the data corresponding to multiple constellations, while *individual* model considers as many instances of each model, as there are constellations – for example, with 2 constellations, the results corresponding to *single model* were obtained by inferring and learning a single common model, while the results corresponding to an *individual* model were obtained from the average performance of C models, each trained and tested only with one of the C deployment constellations.

Moreover, we differentiate between the architectures by adding the $+$ sign, i.e. *individual*$+$ and *single model*$+$, when referring to the second, larger NN architecture. To ensure the fairness of comparison, all considered models were trained with identical settings.

C. Numerical Results

1) Baseline Positioning Performance for a Single gNB Constellation: In the first part of the evaluation, we study the performance of the considered model in a single given constellation, depicted in Fig. 3 with black crosses, to assess the baseline capabilities of the HN as a general positioning model. The corresponding results are depicted in Fig. 4, in terms of the UE localization error cumulative distributions, where the localization error is defined as $\sqrt{(x-\hat{x})^2+(y-\hat{y})^2}$. The proposed HypLoc solution provides the lowest errors with a mean positioning error of 2.81m, followed by the Single model$+$ with a mean positioning error of 3.26m. The Single model, having a significantly lower number of parameters, performs the poorest with a 4.42m mean positioning error. The performance of the single model and individual benchmarks are in this case identical, as only one constellation is so far considered.

The results show that in a gNB-wise static scenario with a single constellation, the application of a HN is justified and more efficient than the traditional models when it comes to the fundamental localization capabilities. This is stemming from HN's ability to dynamically change the model parameters based on the input array, resulting in adaptive model behavior depending on the UE's location.

2) Increasing the Number of gNB Constellations: In the next evaluations, we study the performance behavior of a standard NN, and that of the HypLoc, when increasing the number of possible gNB constellations within the environment. We set up the experiment by randomly creating 10 different constellations with 7 gNBs in each. Then, by iteratively increasing the number of constellations, we train the considered models and evaluate their positioning performance.

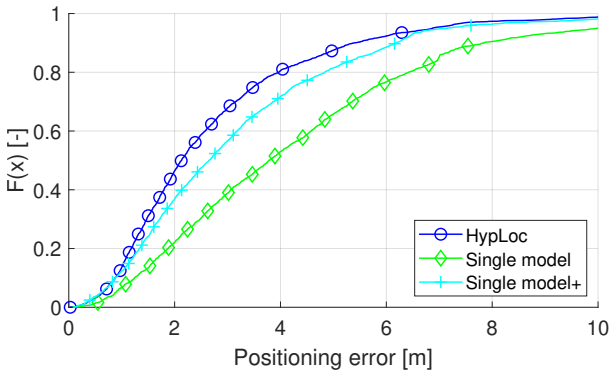

Fig. 4. Distribution of the positioning errors on the test dataset in a single given gNB constellation.

The mean positioning results across considered constellations are depicted in Fig. 5, where the number of gNB constellations increases from 1 to 10. The figure shows consistent results with those in Fig. 4 where only a single constellation was considered. The single model and individual models perform almost the same, while the same is true for the three larger models. The slight differences between the performance of individual and single models at 1 constellation is a direct result of randomness in NN models' initialization and training. When increasing the number of constellations, i.e., the amount of information the models have to adapt to, we can observe the following. The performance of the single model degrades fairly linearly with the increasing number of constellations, with a much greater slope than that of the remaining models. The HypLoc provides lower mean positioning errors than both the single model and the single model$+$ for all numbers of gNB constellations while providing the lowest error across all models at 1 and 2 possible constellations, being further outperformed by the individual$+$ benchmark consisting of up to 10 individually trained models for each constellation. The fluctuations in the observable trends especially with the Individual and Individual$+$ models are caused by variations in signal coverage within the overall Madrid Map region across the different constellations as well as by the inherent randomness within the NN training and initialization processes. The sudden increase in the mean localization error at 6 constellations is caused by an unfortunate selection of the gNBs location, such that a significant part of the overall Madrid map area is left without signal coverage.

To provide a complementary performance measure, the normalized error relative to the performance of the Individual model is depicted in Fig. 6. The figure was obtained by dividing each model's mean positioning error by the error of the Individual model (thus it always yields 1). The data trends are now smoother, and the relative differences between

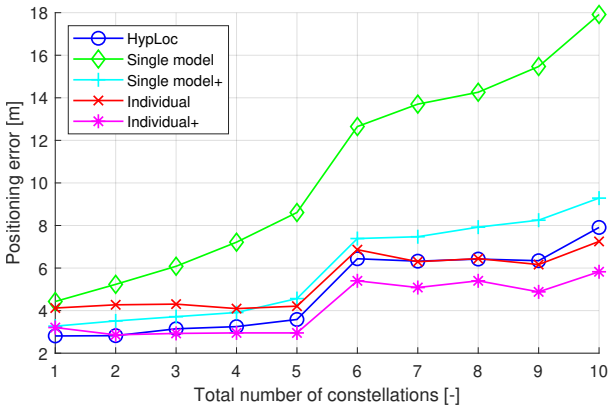

Fig. 5. Mean positioning error across all scenarios ranging from 1 to 10 constellations.

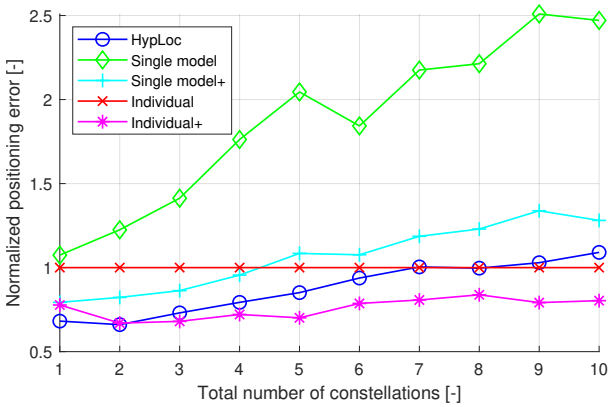

Fig. 6. Mean positioning error across all scenarios normalized to the performance of the Individual model.

the HypLoc's performance and the benchmarks are also better recognizable.

Overall, the numerical results provided along Figs. 5 and 6 indicate the superior capabilities of the HypLoc model over those of the traditional ones when a single neural network entity is operating under a variety of gNB deployments.

3) Generalizing to an Unseen gNB Constellation: In the final and most challenging example, the capabilities of the HypLoc to adapt to a new and previously unseen constellation are evaluated – in the spirit of transfer learning. First, all models are trained in 10 known constellations (corresponding to the results shown in e.g. Fig. 5), being then further re-trained on the data corresponding to the new, unseen gNB constellation. We note that the re-trained Individual and Individual+ models correspond always to the best-performing gNB constellation. Additionally, we manually selected the constellation so that the signal coverage is well-distributed throughout the Madrid map environment, ensuring a feasible localization task.

In general, the rate at which any model can learn from the new data is one crucial characteristic. Fig. 7 provides the comparison of the considered models in terms of their mean positioning error based on the number of epochs trained on, while considering the full set of training samples. The figure shows

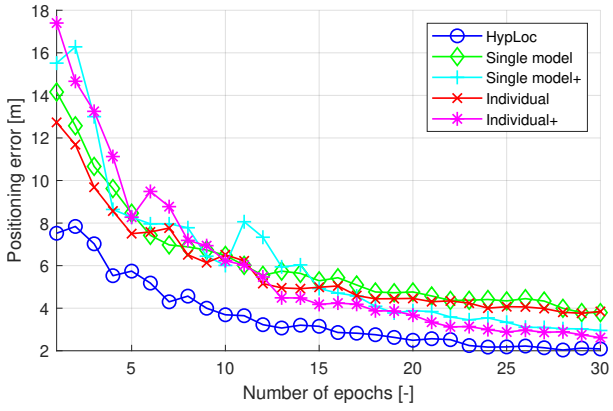

Fig. 7. Mean positioning errors of considered models in a previously unseen gNB deployment scenario based on the re-training progress.

the achieved positioning performance after the first epoch, as without any re-training, the mean positioning errors are still very large. Specifically, without any re-training, the HypLoc provided 140.4m mean positioning error, Single model 187.5m, Single model+ 193.5m, Individual model 167.4m, and Individual+ model 218.5m. Now, for increasing re-training epoch counts, the results in Fig. 7 show that the HypLoc model performs the strongest across the re-training, achieving the lowest positioning errors across all epochs. This is one strong benefit of the proposed scheme compared to the reference solutions.

Furthermore, we study each model's ability to adapt to the new environment with a limited amount of available new data – an important practical aspect related to the costs and efforts in acquiring new data in a new deployment scenario. In the experiment, the data was deliberately limited to 1%, 2%, 5%, 10%, 25%, and 50% of the original training set size, referring to 227 samples, 455 samples, 1138 samples, 2277 samples, 5693 samples, and 11387 samples, respectively. As can be observed through the results in Fig. 8, the proposed HypLoc achieves the lowest positioning error at all data sizes. Additionally, the Single model+ consistently outperforms the Individual+ model, showing that the multiple learned environments positively impact the model's capability to adapt to the previously unknown scenario when data is scarce. Interestingly, with limited data, the Individual model provides better results than its larger counterpart as smaller models have lower training requirements than the complex ones.

D. Further Discussion and Future Work

Reference models: We utilized two reference NN architectures, one with identical positioning pipeline to the proposed HypLoc, and the other, denoted with +, with almost the same number of trainable parameters and thus similar complexity. The performance gap between the smaller model (esp. Single model) shows the direct impact of the HN model on the performance, as HypLoc utilizes the same, "lightweight" positioning pipeline. While the performance gap between the + models and the HypLoc is smaller there is no downside to implementing HypLoc, as utilizing it carries no additional strains in terms of costs or requirements than the standard model.

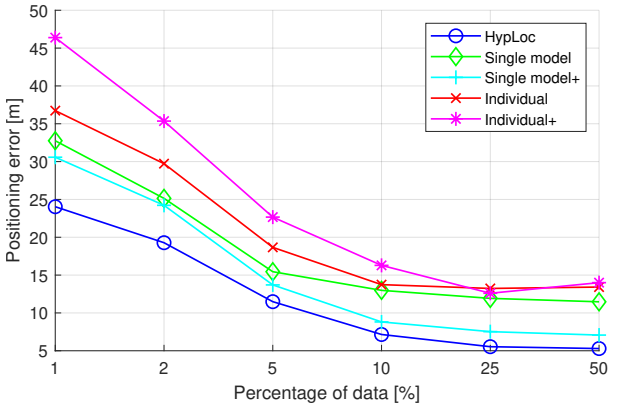

Fig. 8. Mean positioning errors of considered models in a previously unseen gNB deployment scenario based on the available sample scarcity.

Future work: The provided results demonstrate the promising capabilities of the HN and open several research directions. One of the drawbacks of the utilized models is the fixed nature of the implemented architecture in terms of the number of gNBs which is here fixed to 7. Designing a model capable of adjusting not only the weights and biases but the entire architecture, is one challenge that HN may be able to realize. Another major network aspect the HN-based models can potentially enable is alleviating the requirement of model re-training completely. By incorporating enough data and appropriate learning, one may argue that the model can learn to localize the user while the locations of gNBs are arbitrary, which we will strive for in our future work. A natural extension of the current HypLoc model to further improve its positioning performance is incorporating a sequence-processing NN. When processing sequences, HN adapts the weights of the main model in each sample instance and consequently, unrolling the recurrent model results in a series of different models.

V. Conclusion

In this work, we introduced the so-called HypLoc concept referring to a novel HN-based architecture for localization within 5G and beyond networks. After shortly reviewing the basics of HNs and available 5G positioning measurements, the paper presented the architectural design, hyperparametrization, and training approach leading to the proposed HypLoc concept. The ray-tracing-based numerical evaluations showed the superior positioning performance of the proposed model in an urban deployment scenario, to localize vehicular terminals, achieving almost 20% lower mean positioning error than a traditional NN model with the same number of parameters. Furthermore, we investigated the capabilities of HypLoc to generalize across multiple environments by increasing the number of possible gNB constellations – again leading to consistently improved results compared to traditional models. Finally, the advantages of parametrizing the model using a HN were also shown when adapting the model to a previously unseen gNB deployment constellation where the HypLoc model adapts faster and with less training data than the other existing NN models.

References

[1] K. B. Letaief *et al.*, "The roadmap to 6G: AI empowered wireless networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[2] A. Jagannath, J. Jagannath, and T. Melodia, "Redefining Wireless Communication for 6G: Signal Processing Meets Deep Learning with Deep Unfolding," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 6, pp. 528–536, 2021.

[3] J. Zhang, C.-K. Wen, and S. Jin, "Adaptive MIMO detector based on hypernetwork: Design, simulation, and experimental test," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 65–81, 2021.

[4] G. Liu *et al.*, "A Hypernetwork Based Framework for Non-Stationary Channel Prediction," *IEEE Transactions on Vehicular Technology*, 2024.

[5] S. Xie *et al.*, "Deep Learning-Based Adaptive Joint Source-Channel Coding using Hypernetworks," *arXiv preprint*, 2024.

[6] W.-C. Tsai *et al.*, "Low-Complexity Compressive Channel Estimation for IRS-Aided mmWave Systems with Hypernetwork-Assisted LAMP Network," *IEEE Communications Letters*, vol. 26, no. 8, pp. 1883–1887, 2022.

[7] R. Klus *et al.*, "Machine Learning Based NLOS Radio Positioning in Beamforming Networks," in *Proc. IEEE SPAWC*, 2022, pp. 1–5.

[8] C. De Lima *et al.*, "Convergent communication, sensing and localization in 6G systems: An overview of technologies, opportunities and challenges," *IEEE Access*, vol. 9, pp. 26 902–26 925, 2021.

[9] R. Mendrzik *et al.*, "Enabling Situational Awareness in Millimeter Wave Massive MIMO Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 1196–1211, 2019.

[10] M. M. Butt, A. Pantelidou, and I. Z. Kovács, "ML-assisted UE positioning: Performance analysis and 5G architecture enhancements," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 377–388, 2021.

[11] D. Lynch *et al.*, "Localisation in wireless networks using deep bidirectional recurrent neural networks," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 2020, pp. 1–8.

[12] Z. Dai *et al.*, "DeepAoANet: Learning Angle of Arrival From Software Defined Radios With Deep Neural Networks," *IEEE Access*, vol. 10, pp. 3164–3176, 2022.

[13] Y. Assayag *et al.*, "Indoor positioning system using synthetic training and data fusion," *IEEE Access*, vol. 9, pp. 115 687–115 699, 2021.

[14] K. Gao *et al.*, "Toward 5G NR high-precision indoor positioning via channel frequency response: A new paradigm and dataset generation method," *IEEE Journal on Selected Areas in Communications*, 2022.

[15] V. K. Chauhan *et al.*, "A Brief Review of Hypernetworks in Deep Learning," *arXiv preprint*, 2023.

[16] D. Ha, A. Dai, and Q. V. Le, "HyperNetworks," *arXiv preprint*, 2016.

[17] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep hypernetwork-based MIMO detection," in *Proc. IEEE SPAWC*, 2020, pp. 1–5.

[18] C.-J. Wang *et al.*, "Phase retrieval using expectation consistent signal recovery algorithm based on hypernetwork," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5770–5783, 2021.

[19] K. Pratik *et al.*, "Neural augmentation of Kalman filter with hypernetwork for channel tracking," in *Proc. IEEE GLOBECOM*, 2021, pp. 1–6.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT press, 2016.

[21] P. Ferrand, A. Decurninge, and M. Guillaud, "DNN-based localization from channel estimates: Feature design and experimental results," in *Proc. IEEE GLOBECOM*, 2020, pp. 1–6.

[22] 3GPP, "Physical layer measurements," Technical Report (TR) 38.215, V16.4.0, Jan. 2021.

[23] 3GPP, "Stage 2 functional specification of User Equipment (UE) positioning in NG-RAN," Technical Report (TR) 38.305, V16.7.0, Dec. 2021.

[24] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint*, 2016.

[25] Remcom. Wireless InSite - 3D Wireless Prediction Software. Accessed: Jan 27, 2021). [Online]. Available: https://www.remcom.com/wireless-insite-em-propagation-software

[26] A. Rauch *et al.*, "Fast algorithm for radio propagation modeling in realistic 3-D urban environment," *Advances in Radio Science*, vol. 13, pp. 169–173, 11 2015.

[27] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," Technical Report (TR) 38.901, V17.0.0, March 2022.