Towards Automated and Interpretable Pathloss Approximation Methods

Ahmad Anaqreh*, Shih-Kai Chou, Irina Barašin, Carolina Fortuna

Jožef Stefan Institute, Slovenia

Abstract

Modeling propagation path loss is crucial for optimizing next-generation wireless communication systems, including 5G and beyond. This work explores the use of Deep Symbolic Regression and Kolmogorov-Arnold Networks as innovative methods for approximating path loss models such as Alpha-Beta-Gamma and Close-In which are commonly used in urban micro- and macro-cellular scenarios. By integrating the predictive power of machine learning with the symbolic approaches, these methods achieve high accuracy approximation across a wide range of frequencies and propagation conditions. Through neural-guided symbolic regression and interpretable architectures, this work demonstrates how these approaches can simplify path loss modeling while maintaining robust performance. Validating these methods highlights their potential to effectively approximate path loss models in wireless communication systems.

Introduction

In recent years, the rapid evolution of cellular networks has revolutionized the modern digital landscape, carrying massive volumes of traffic on a global scale. Several technologies and architectures have been introduced to the beyond 5G era to boost the networks performance on throughput or latency, such as millimeter Wave (mmWave) (Wang et al. 2018), beamforming (Van Veen and Buckley 1988), and O-RAN (Polese et al. 2023). Particularly, mmWave technologies involve higher carrier frequencies, which provide enormous bandwidth potential but suffer from significant path loss due to the short wavelength and limited diffraction capabilities (Uwaechia and Mahyuddin 2020). This makes the characterization of channels critically important, as accurate channel information enables the effective deployment of mmWave networks by compensating for these propagation challenges. Moreover, beamforming relies on precise Channel State Information (CSI) to steer signals toward the intended receiver, further maximizing signal strength and minimizing interference (Pourkabirian et al. 2024). Without accurate channel characterization, beamforming would be inefficient and could lead to suboptimal network performance. Furthermore, O-RAN, as an emerging architecture, leverages Artificial intelligent (AI)/ Machine learning (ML) to enhance service diversity and optimize network performance, including channel characterization, by moving away from traditional stochastic models such as the Alpha-Beta-Gamma (ABG) and Close-In (CI) models, it can offer a data-driven approach to network optimization.

These traditional models have long been considered essential for predicting path loss in 5G and other modern wireless systems. These models utilize parameter fitting techniques to describe propagation characteristics over varying distances, frequencies, and environmental scenarios (Sun et al. 2016a). Moreover, these models require significant computational effort to retrieve their parameter values, particularly when applied to complex urban settings with varying propagation environments. As wireless communication systems continue to expand, there is an increasing need for methodologies that enhance accuracy and provide interpretable and computationally efficient solutions for modeling propagation path loss. Advanced machine learning techniques, such as Deep Symbolic Regression (DSR) (Petersen et al. 2019) and Kolmogorov-Arnold Networks (KANs) (Liu et al. 2024), have emerged as promising approaches to address these challenges.

DSR is a methodology designed to extract interpretable and straightforward mathematical expressions from given data. By using the power of neural networks in combination with reinforcement learning, DSR explores the vast space of potential equations with remarkable efficiency. This hybrid approach represents a significant advancement over traditional symbolic regression methods, such as Genetic Programming (GP) (Koza et al. 1992), which rely on evolutionary mechanisms. DSR, in contrast, employs a risk-seeking policy gradient strategy that aims to discover high-quality expressions instead of simply optimizing average outcomes. The ability to recover complex relationships, even in noisy or high-dimensional data, is possible through integrating neural networks, which guide the search process and reinforcement learning that dynamically refine it, making DSR a valuable tool across diverse applications, such as engineering optimization, and predictive modeling.

KANs represent a novel neural network architecture designed to model complex relationships in data while maintaining a substantial focus on interpretability and transparency. Drawing inspiration from the Kolmogorov-Arnold

*Email: alanaqrehahmad88@gmail.com

representation theorem, which states that any multivariate continuous function can be defined as a sum of univariate functions, KANs decompose data into simpler components. This approach is particularly valuable in applications where understanding the model's behavior is as critical as its predictive performance. Unlike traditional neural networks, which often function as "black boxes", KANs apply activation functions along the edges connecting nodes rather than at the nodes themselves, enabling a clearer mapping between inputs and outputs, and enhancing interpretability.

The contribution of this paper consists of investigating the feasibility of interpretable machine learning models to perform automated and interpretable path loss approximation. Our results show the superiority of KANs in approximating selected path loss models. While these architectures have been considered for other scientific areas such as physics, mechanics, and medicine, to the best of our knowledge this is the first attempt towards exploration in wireless communication networks.

Problem Statement

We aim to approximate two path loss models discussed in (Sun et al. 2016a; MacCartney et al. 2013; Piersanti, Annoni, and Cassioli 2012; Andersen, Rappaport, and Yoshida 1995), namely the ABG and CI models. These models are widely adopted in wireless communication systems. Specifically, they are designed to describe and predict the behavior of signal propagating in 5G urban micro- and macro-cellular scenarios, incorporating factors such as frequency, distance, and shadowing factor.

The ABG model is defined by the equation:

$$PL^{\text{ABG}}(f,d) = 10\alpha \log_{10}\left(\frac{d}{1\,\text{m}}\right) + \beta + 10\gamma \log_{10}\left(\frac{f}{1\,\text{GHz}}\right) + \chi^{\text{ABG}} \quad (1)$$

where $PL^{\text{ABG}}(f,d)$ is the path loss in dB, α and γ are coefficients representing dependence on distance (d), frequency (f), respectively. β is an optimized offset value for path loss, and χ^{ABG} represents the standard deviation of shadow fading.

The CI model is described by the equation:

$$PL^{\text{CI}}(f,d) = FSPL(f, 1\,\text{m}) + 10n \log_{10}(d) + \chi^{\text{CI}} \quad (2)$$

where n is the path loss exponent (PLE), d is the 3D transmitter-receiver distance, $FSPL(f, 1\,\text{m})$ is the free-space path loss at 1 meter, and χ^{CI} represents the shadow fading. $FSPL(f, 1\,\text{m})$ is computed as:

$$FSPL(f, 1\,\text{m}) = 20 \log_{10}\left(\frac{4\pi f}{c}\right) \quad (3)$$

where c is the speed of light.

In this paper, we follow the typical methodology from the general symbolic regression community (Makke and Chawla 2024) to find architectures that best approximate the theoretical values computed by the path loss model. Exact symbolic match as well as domain-specific interpretability and performance aspects are left for future work.

Methods

To approximate the selected path loss models, we identify three interpretable candidate methods capable of learning a symbolic representation of the data at hand: deep symbolic regression, symbolic regression with evolutionary computation, and the recently introduced Kolmogorov-Arnold Networks.

Deep Symbolic Regression

The core of the DSR (Petersen et al. 2019) lies in representing mathematical expressions as sequences, developing an autoregressive model to generate these expressions, and utilizing a risk-seeking policy gradient approach to train the model for generating more precise and well-fitting expressions.

The sequence generator represents a parameterized distribution over mathematical expressions, $p(\tau \mid \theta)$. The model is typically designed to ensure that the likelihood of an expression is computationally tractable for the parameters θ, enabling the use of backpropagation with a differentiable loss function. A common implementation is a recurrent neural network (RNN), where the likelihood of the i-th token (τ_i) is conditionally independent of other tokens, given the initial ones ($\tau_1, \ldots, \tau_{i-1}$). This is expressed as:

$$p(\tau_i \mid \tau_{j \neq i}, \theta) = p(\tau_i \mid \tau_{j<i}, \theta) \quad (4)$$

The sequence generator is generally trained using reinforcement learning or similar techniques. From this perspective, the sequence generator works as a reinforcement learning policy to be optimized. This involves sampling a batch of N expressions $\mathcal{T}$, evaluating each expression using a reward function $R(\tau)$, and applying gradient descent to minimize a loss function. In this work, we utilized three approaches discussed in (Petersen et al. 2019) for training the RNN:

- **Risk-Seeking Policy Gradient (RSPG):** Proposed in (Petersen et al. 2019), focuses on optimizing the best-case reward rather than the average. The loss function is defined as:

$$\mathcal{L}(\theta) = \frac{1}{\epsilon|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \big(R(\tau) - \tilde{R}_\epsilon\big) \nabla_\theta \log p(\tau \mid \theta) \mathbf{1}_{R(\tau) > \tilde{R}_\epsilon} \quad (5)$$

  where ϵ is a hyperparameter controlling the level of risk-seeking, $\tilde{R}_\epsilon$ is the empirical $(1-\epsilon)$ quantile of the rewards in $\mathcal{T}$, and $\mathbf{1}$ is an indicator function.

- **Vanilla Policy Gradient (VPG):** This method uses the REINFORCE algorithm (Williams 1992), where training is performed over the batch $\mathcal{T}$ with the loss function:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \big(R(\tau) - b\big) \nabla_\theta \log p(\tau \mid \theta) \quad (6)$$

  where b is a baseline term, such as an exponentially weighted moving average (EWMA) of rewards.

- **Priority Queue Training (PQT):** Introduced by Abolafia (Abolafia et al. 2018), this non-reinforcement learning method emphasizes best-case performance. Samples

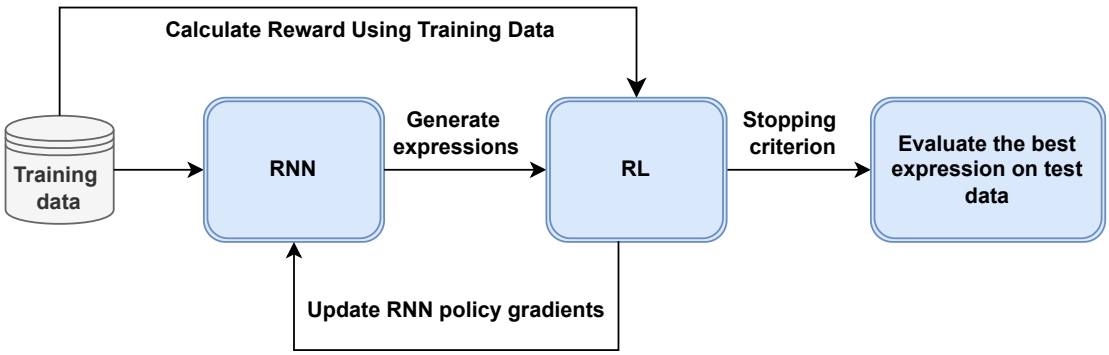

Figure 1: The flow diagram of Deep Symbolic Regression.

from each batch are stored in a maximum reward priority queue (MRPQ). Training is then performed on these stored samples using a supervised learning objective:

$$\mathcal{L}(\theta) = \frac{1}{k} \sum_{\tau \in \text{MRPQ}} \nabla_\theta \log p(\tau \mid \theta) \tag{7}$$

where k is the size of the MRPQ.

For a pre-order traversal τ and a dataset of (X, y) pairs of size N, where $X \in \mathbb{R}^n$ and $y \in \mathbb{R}$, the normalized root-mean-square error (NRMSE) is defined as:

$$\text{NRMSE}(\tau) = \frac{1}{\sigma_y} \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - f(X_i))^2} \tag{8}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the mathematical expression instantiated from τ, and σ_y is the standard deviation of y. Thus, The reward function is described as:

$$R(\tau) = \frac{1}{1 + \text{NRMSE}(\tau)} \tag{9}$$

In DSR, it is possible to implement a set of constraints to reduce the search space, ensuring the generation of meaningful expressions while maintaining computational efficiency. The following constraints were specifically applied:

1. **Expression Length Limits:** Expressions were restricted to a minimum length of 4 to avoid trivial solutions and a maximum length of 30 to ensure interpretability.
2. **Avoiding Redundant Constants:** Operators were restricted to prevent all their children from being constants, as this would produce expressions that simplify to a single constant.
3. **Unary Operator Consistency:** Unary operators were disallowed from having their inverse as a child (e.g., $\log(\exp(x))$), as such expressions are redundant.
4. **Trigonometric Operator Composition:** Trigonometric operators were prevented from having other trigonometric operators as descendants (e.g., $\sin(1+\cos(x))$). While such compositions are mathematically valid, they occasionally appear and cause unnecessary complexity.

The details of the flow diagram used for approximating the path loss ABG and CI models in Eq. 1 and 2 with the DSR method are illustrated in Figure 1. As depicted in the figure, Recurrent Neural Network (RNN) generates symbolic expressions from the training data, adhering to any predefined constraints. The role of Reinforcement Learning (RL) is to guide the RNN by assigning a reward to each generated expression, based on its performance on the training data. This reward measures how well the expression fits the data, balancing predictive accuracy and model complexity. Using policy gradient methods, the RL algorithm updates the RNN parameters to progressively generate higher-quality expressions. The process is iterative, continuing until a stopping criterion, such as convergence or a performance threshold, is reached. Finally, the best expression is validated on test data to confirm its generalization capability.

Genetic Programming

GP (Koza et al. 1992) is an evolutionary computation method that optimizes mathematical expressions represented as tree structures which are mathematically described as:

$$T = \{N, L, F\} \tag{10}$$

where:

- N is the set of nodes.
- L is the set of leaves (constants or variables).
- F is the set of operators (e.g., $+, -, \times, \div, \log, ...$).

Each expression, known as an "individual", operates as a potential solution.

GP process begins with an initial population, often generated randomly, and iteratively improves it through a series of operations inspired by natural evolution. These operations include mutation, which applies random changes to individuals, such as replacing subtrees with newly generated random subtrees, crossover, which exchanges subtrees between two individuals to create offspring, and selection, which determines which individuals remain in the next generation based on their fitness. Fitness is calculated using a predefined function, which evaluates the quality of a solution.

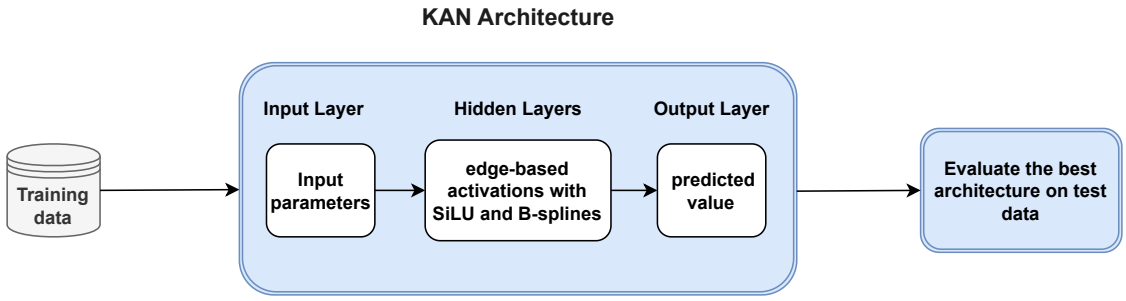

Figure 2: The flow diagram of KANs.

In each generation, individuals are selected for reproduction, modified by mutation or crossover, and replaced to form the next population. A common selection strategy is tournament selection, where a subset of individuals is sampled, and the best among them is chosen to remain. This iterative process continues until a termination criteria, such as a maximum number of generations or a target fitness, is satisfied.

Kolmogorov-Arnold Networks

KANs have been recently introduced as an alternative to MLP-based architectures and shown to perform well on small scientific tasks while also being interpretable (Vaca-Rubio et al. 2024).

The structure of a KAN can be represented as $[n_1, \ldots, n_{L+1}]$, where L signifies the total number of layers in the KAN. A deeper KAN can be thus formulated through the composition of L layers as:

$$Y = \text{KAN}(X) = (\Phi_L \circ \Phi_{L-1} \circ \cdots \circ \Phi_1)X \tag{11}$$

Each layer of a KAN is represented by a matrix where each entry is an activation function. If there is a layer with d_{in} nodes and its neighboring layer with d_{out} nodes, the layer can be represented as a $d_{in} \times d_{out}$ matrix of activation functions.

$$\Phi = \{\phi_{q,p}\},\ p = 1, 2, \ldots, d_{in},\ q = 1, 2, \ldots, d_{out} \tag{12}$$

Unlike traditional MLPs, where activation functions are applied at the nodes themselves, KAN places them at the edges between the nodes. KAN employs the SiLU activation function in combination with B-splines to enhance its expressiveness. This setup allows the edges to control the transformations between layers, while the nodes perform simple summation operations. A B-spline of order k requires $G + k$ basis functions to represent the spline over the grid. For each input (node in a layer), evaluating a B-spline of order k thus involves computing $G+k-1$ basis functions and performing a weighted sum with the corresponding control points.

$$\text{spline}(x) = \sum_{i=0}^{G+k-1} c_i B_i(x) \tag{13}$$

The flow diagram illustrating the approximation of the path loss ABG and CI models, as described in Eqs. 1 and 2, using KANs is shown in Figure 2. This workflow involves training data, the KAN architecture, and evaluation of test data. However, during the training, several configurations regarding layers and grid size were considered for the KANs architecture. Using a manual trial and error approach, the highest performance for both models is reported here.

Figure 3 illustrates the KAN architecture that was evaluated for the ABG model, with a three-layer design [6, 6, 1]. The first layer includes 6 input nodes, corresponding to the parameters $\alpha, \gamma, \beta, f, d, \chi$ from Eq. 1, we chose the second layer having the same number of nodes and the output layer consists of a single node representing the predicted value of the function. The network utilizes a structure with grid=10, allowing for a high-resolution representation of the feature space, and k=3 defines the degree of the splines. The model is optimized using the LBFGS method over 500 training steps.

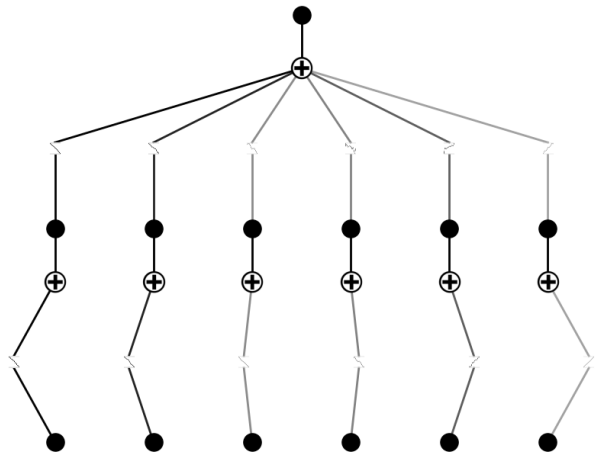

Figure 3: [6,6,1] KAN architecture for ABG model.

Figure 4 displays the KAN architecture designed for the CI model, also a three-layer configuration with [4, 4, 1]. The input layer comprises 4 nodes, representing the parameters f, d, χ, n from Eq. 2, the middle layer includes the same number of nodes while the output layer includes a single node providing the predicted value of the function. Similar

to the ABG model, the network is configured with grid=10, k=3, and is trained over 500 steps.

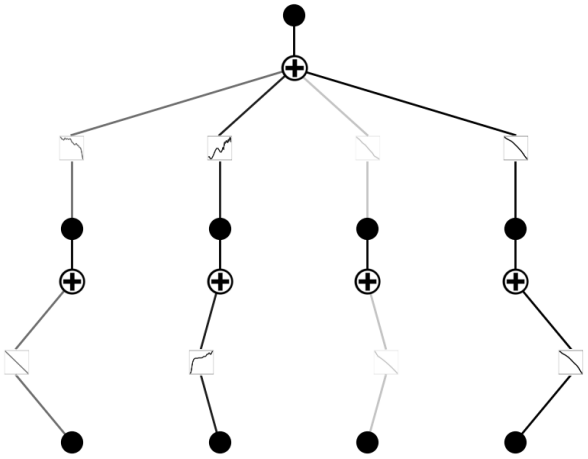

Figure 4: [4,4,1] KAN architecture for CI model.

Training and Evaluation

To learn to approximate the path loss methods given by Eq. 1 and 2, we used the respective formulas to create training data for the models discussed above. The used parameter ranges are presented in Table 1. These ranges are derived from authentic data in studies that comprehensively analyze various scenarios and frequency ranges, highlighting the applicability of the ABG and CI models in diverse contexts (Sun et al. 2016a,b; Cheng, Kim, and Zajić 2017).

Parameter	ABG	CI
$\boldsymbol{\alpha}$	$[0.1, 2.5]$	-
$\boldsymbol{\beta}$	$[-10, -1]$	-
$\boldsymbol{\gamma}$	$[0, 2]$	-
$\boldsymbol{f}$	$[2, 73.5]$	$[2, 73.5]$
$\boldsymbol{d}$	$[1, 500]$	$[1, 500]$
$\boldsymbol{\chi}$	$[4, 12]$	$[4, 12]$
$\boldsymbol{n}$	-	$[2, 6]$

Table 1: The parameters and their corresponding values for ABG and CI models.

We created a dataset consisting of 1,000 instances for DSR, KANs, and GP with 800 instances assigned for training and 200 for testing. The performance of the approaches was evaluated using four key metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2).

Results

The performance comparison of the different methods on ABG model is presented in Table 2. Among these methods, KANs stand out as the most accurate and reliable, achieving the lowest MAE (0.02) and MSE (0.001), indicating minimal prediction errors. Additionally, KANs' MAPE of 6.49 reflects a low percentage error, and its R^2 value of 0.98 demonstrates that it explains nearly all the variance in the data. In comparison, RSPG shows moderate performance with an R^2 of 0.82 and MAPE of 11.50 but falls short in accuracy with higher MAE (6.53) and MSE (20.92). VPG and PQT perform less effectively, with high MAPE values (59.11 and 79.77, respectively) and weaker R^2 scores (0.77 and 0.90), indicating significant prediction errors and reduced reliability. GP, while achieving moderate MAE (9.59) and MSE (10.16), suffers from a high MAPE of 21.17 and an R^2 of only 0.16, indicating poor overall performance and limited ability to explain the variability in the data.

Algorithms	MAE	MSE	MAPE	R^2
KANs	0.02	0.001	6.49	0.98
RSPG	6.53	20.92	11.50	0.82
VPG	8.30	90.45	59.11	0.77
PQT	9.31	164.37	79.77	0.90
GP	9.59	10.16	21.17	0.16

Table 2: Performance comparison on ABG model.

For CI model, the results are shown in Table 3. KANs attain superior performance compared to other methods, achieving the lowest MAE (0.01), MSE (0.0002), and MAPE (1.77), with an exceptional R^2 value of 0.99, demonstrating good alignment with observed data. RSPG shows relatively strong performance, with an R^2 of 0.92 and moderate MAPE (10.58), but it lags behind KANs due to higher MAE (20.01) and MSE (168.19). VPG delivers weaker results, with higher MAE (23.34), MSE (233.22), and MAPE (13.31), and a lower R^2 of 0.67, indicating reduced reliability. PQT shows high values in error metrics, (e.g., MAE = 28.75, MSE = 1472.88, and MAPE = 61.45), though it achieves a moderate R^2 of 0.83. Lastly, GP performs the worst, with the highest MAE (48.05), MSE (3292.78), and MAPE (88.25), along with a poor R^2 value of 0.49, indicating limited predictive ability.

Algorithms	MAE	MSE	MAPE	R^2
KANs	0.01	0.0002	1.77	0.99
RSPG	20.01	168.19	10.58	0.92
VPG	23.34	233.22	13.31	0.67
PQT	28.75	1472.88	61.45	0.83
GP	48.05	3292.78	88.25	0.49

Table 3: Performance comparison on CI model.

Overall, KANs outperformed other methods across both ABG and CI models, achieving the lowest error metrics and the highest R^2 values, demonstrating strong predictive reliability. Other methods showed moderate to poor performance, with GP having the highest error values and weakest predictive accuracy.

Conclusion

This paper emphasizes the potential of advanced machine learning methods, such as DSR and KANs, for approximating propagation path loss models. The results demonstrate that these methods offer a novel approach for achieving remarkable accuracy in estimating propagation path loss values, establishing a strong foundation for further research into data-driven and symbolic methodologies for modeling wireless channels. In this paper, we utilize authentic datasets to address the complexities of practical telecommunications scenarios, ensuring the relevance and effectiveness of the methods in real-world applications. As a key objective for future work, we aim to derive interpretable mathematical expressions that provide better insight into path loss equations while maintaining robust performance.

Acknowledgments

This work was supported in part by the European Commission NANCY project (No.101096456) and by the Slovenian Research Agency under grants P2-0016 and J2-50071.

References

Abolafia, D. A.; Norouzi, M.; Shen, J.; Zhao, R.; and Le, Q. V. 2018. Neural program synthesis with priority queue training. *arXiv preprint arXiv:1801.03526*.

Andersen, J. B.; Rappaport, T. S.; and Yoshida, S. 1995. Propagation measurements and models for wireless communications channels. *IEEE Communications magazine*, 33(1): 42–49.

Cheng, C.-L.; Kim, S.; and Zajić, A. 2017. Comparison of path loss models for indoor 30 GHz, 140 GHz, and 300 GHz channels. In *2017 11th European conference on antennas and propagation (EUCAP)*, 716–720. IEEE.

Koza, J. R.; et al. 1992. Evolution of subsumption using genetic programming. In *Proceedings of the first European conference on artificial life*, 110–119. MIT Press Cambridge, MA, USA.

Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T. Y.; and Tegmark, M. 2024. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.

MacCartney, G. R.; Zhang, J.; Nie, S.; and Rappaport, T. S. 2013. Path loss models for 5G millimeter wave propagation channels in urban microcells. In *2013 IEEE global communications conference (GLOBECOM)*, 3948–3953. IEEE.

Makke, N.; and Chawla, S. 2024. Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review*, 57(1): 2.

Petersen, B. K.; Landajuela, M.; Mundhenk, T. N.; Santiago, C. P.; Kim, S. K.; and Kim, J. T. 2019. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. *arXiv preprint arXiv:1912.04871*.

Piersanti, S.; Annoni, L. A.; and Cassioli, D. 2012. Millimeter waves channel measurements and path loss models. In *2012 IEEE International Conference on Communications (ICC)*, 4552–4556. IEEE.

Polese, M.; Bonati, L.; D'Oro, S.; Basagni, S.; and Melodia, T. 2023. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Communications Surveys Tutorials*, 25(2): 1376–1411.

Pourkabirian, A.; Koushki, F.; Torabian, M.; and Rezvani, M. H. 2024. A robust beamforming with antenna selection approach for dense antenna arrays under limited CSI. *Physical Communication*, 65: 102371.

Sun, S.; Rappaport, T. S.; Rangan, S.; Thomas, T. A.; Ghosh, A.; Kovacs, I. Z.; Rodriguez, I.; Koymen, O.; Partyka, A.; and Jarvelainen, J. 2016a. Propagation path loss models for 5G urban micro-and macro-cellular scenarios. In *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 1–6. IEEE.

Sun, S.; Rappaport, T. S.; Thomas, T. A.; Ghosh, A.; Nguyen, H. C.; Kovacs, I. Z.; Rodriguez, I.; Koymen, O.; and Partyka, A. 2016b. Investigation of prediction accuracy, sensitivity, and parameter stability of large-scale propagation path loss models for 5G wireless communications. *IEEE transactions on vehicular technology*, 65(5): 2843–2860.

Uwaechia, A. N.; and Mahyuddin, N. M. 2020. A Comprehensive Survey on Millimeter Wave Communications for Fifth-Generation Wireless Networks: Feasibility and Challenges. *IEEE Access*, 8: 62367–62414.

Vaca-Rubio, C. J.; Blanco, L.; Pereira, R.; and Caus, M. 2024. Kolmogorov-arnold networks (kans) for time series analysis. *arXiv preprint arXiv:2405.08790*.

Van Veen, B.; and Buckley, K. 1988. Beamforming: a versatile approach to spatial filtering. *IEEE ASSP Magazine*, 5(2): 4–24.

Wang, X.; Kong, L.; Kong, F.; Qiu, F.; Xia, M.; Arnon, S.; and Chen, G. 2018. Millimeter Wave Communication: A Comprehensive Survey. *IEEE Communications Surveys Tutorials*, 20(3): 1616–1653.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8: 229–256.