KANsformer for Scalable Beamforming

Xinke Xie, Yang Lu, *Member, IEEE*, Chong-Yung Chi, *Life Fellow, IEEE*, Wei Chen, *Senior Member, IEEE*, Bo Ai, *Fellow, IEEE*, and Dusit Niyato, *Fellow, IEEE*

***Abstract*—This paper proposes an unsupervised deep-learning (DL) approach by integrating Transformer and Kolmogorov–Arnold networks (KAN) termed KANsformer to realize scalable beamforming for mobile communication systems. Specifically, we consider a classic multi-input single-output energy efficiency maximization problem subject to the total power budget. The proposed KANsformer first extracts hidden features via a multi-head self-attention mechanism and then reads out the desired beamforming design via KAN. Numerical results are provided to evaluate the KANsformer in terms of generalization performance, transfer learning and ablation experiment. Overall, the KANsformer outperforms the existing benchmark DL approaches, and is adaptable to the variation in the number of mobile users with real-time and near-optimal inference.**

***Index Terms*—Transformer, KAN, beamforming, energy efficiency.**

I. Introduction

Deep learning (DL) has revolutionized a wide range of application fields and achieved unprecedented success in tasks such as image recognition and natural language processing. Its ability to automatically extract high-level features from raw data enables deep neural networks to outperform traditional machine learning methods in complex problem domains. Recently, the DL-enabled designs for wireless networks have emerged as a hot research topic [1]. Some researchers attempted to apply multi-layer perceptrons (MLP) [2], convolutional neural networks (CNN) [3] and graph neural networks (GNN) [4] to deal with the power allocation and signal processing problems in wireless networks. Overall, the DL models can be trained to achieve close performance to traditional convex optimization (CVXopt)-based approaches but with a much faster inference speed. How to further improve the learning performance remains an open issue for DL-enabled wireless optimization.

Typically, task-oriented DL requires dedicated models for wireless networks. One promising way is to follow the "encoder-decoder" paradigm, where the encoder extracts features over the wireless networks while the decoder maps the extracted features to desired transmit design. Recent works have paid great attention to the design of encoder. Particularly, the GNN shows good scalability and generalization performance by exploiting the graph topology of wireless networks [5]. In [6]–[8], the GNN was adopted as the encoder to develop the solution approaches for energy efficiency (EE) maximization, sum-rate maximization and max-min rate, respectively, for multi-user multi-input single-output (MISO) networks, all of which were scalable[1] to the number of users. In [10], a GNN based model was trained via unsupervised learning to solve the outage-constrained EE maximization problem. The GNNs in [6]–[8], [10] all leveraged the multi-head attention mechanism also known as the graph attention networks (GAT) to enhance the feature extraction, especially for inter-user interference. Similarly, Transformer is also built upon the self-attention mechanism and is popular for its highly predictive performance [11]. In [12], a Transformer and a weighted A^* based algorithm were proposed to plan the unmanned aerial vehicle trajectory for age-of-information minimization, which outperformed traditional algorithms numerically. However, the above works on wireless networks all adopted MLP as decoder. Recently, Kolmogorov-Arnold network (KAN) has been proposed as a promising alternative to MLP with superior performance in terms of accuracy and interpretability, which is strongly theoretically guaranteed by the Kolmogorov-Arnold Representation Theorem [13].

To the best of our knowledge, the integration of Transformer and KAN has not yet been applied to the beamforming design. Such an integration can inherit the feature extracting capability of Transformer and the feature parsing capability of KAN, thus enhancing the "encoder-decoder" in both aspects. In this paper, we formulate the classic EE maximization problem for MISO networks [14]. We then propose an approach integrating Transformer and KAN termed KANsformer, which utilizes the multi-head self-attention mechanism to extract hidden features among interference links and KAN to map the extracted features to the desired beamforming design. The KANsformer is trained via unsupervised learning and a scale function guarantees feasible solution. Via parameter sharing, the KANsformer is scalable to the number of users, which enhances the KANsformer to directly handle variant problem scales without retraining. Numerical results indicate that the KANsformer outperforms existing DL models and approaches the CVXopt-based solution accuracy with millisecond-level inference time. The major performance gain is contributed by

This work was supported in part by Beijing Natural Science Foundation under Grant L221010 and L242086, in part by Beijing Nova Program under Grant 20230484407, and in part by National Natural Science Foundation of China (NSFC) under Grant U2468201 and 62221001. The work of Chong-Yung Chi was supported by National Science and Technology Council (NSTC) under Grant NSTC 113-2221-E-007-097. *(Corresponding author: Yang Lu.)*

Xinke Xie and Yang Lu are with the State Key Laboratory of Advanced Rail Autonomous Operation, and also with the School of Computer Science and Technology, Beijing Jiaotong University, Beijing 100044, China (e-mail: 21261022@bjtu.edu.cn, yanglu@bjtu.edu.cn).

Chong-Yung Chi is with the Institute of Communications Engineering, Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail:cychi@ee.nthu.edu.tw).

Wei Chen and Bo Ai are with the School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: weich@bjtu.edu.cn, boai@bjtu.edu.cn).

Dusit Niyato is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: dniyato@ntu.edu.sg).

[1]"Scalable" represents the ability of DL models to generalize to different problem scales [9]. Hence, the input-output dimensions can be invariant with the number of users.

the KAN via ablation experiment. Besides, we validate the scalability of KANsformer, which can be further enhanced by transfer learning at the expense of little training cost.

The rest of this paper is organized as follows. Section II gives the system model and problem formulation. Section III presents the structure of KANsformer. Section IV provides numerical results. Finally, Section V concludes the paper with future research directions.

II. System Model and Problem Formulation

Consider a downlink MISO network, where one N_{T}-antenna transmitter intends to serve K single-antenna mobile users (MUs) over a common spectral band. We use $\mathcal{K} \triangleq \{1, 2, ..., K\}$ to denote the index set of the MUs.

Denote the symbol for the k-th MU and the corresponding beamforming vector as s_k and $\mathbf{w}_k \in \mathbb{C}^{N_{\mathrm{T}}}$, respectively. The received signal at the k-th MU is given by

$$\mathrm{y}_k = \mathbf{h}_k^H \mathbf{w}_k s_k + \sum\nolimits_{i \neq k}^{K} \mathbf{h}_k^H \mathbf{w}_i s_i + n_k, \tag{1}$$

where $\mathbf{h}_k \in \mathbb{C}^{N_{\mathrm{T}}}$ denotes the channel state information (CSI) of the k-th transmitter-MU link, and $n_k \sim \mathcal{CN}(0, \sigma_k^2)$ denotes the additive white Gaussian noise (AWGN) at the k-th MU. Without loss of generality, it is assumed that $\mathbb{E}\{|s_k|^2\} = 1$ ($\forall k \in \mathcal{K}$). Then, the achievable rate at the k-th MU is expressed as

$$R_k\left(\{\mathbf{w}_i\}\right) = \log_2\left(1 + \frac{\left|\mathbf{h}_k^H \mathbf{w}_k\right|^2}{\sum_{i=1, i\neq k}^{K} \left|\mathbf{h}_k^H \mathbf{w}_i\right|^2 + \sigma_k^2}\right), \tag{2}$$

where $\{\mathbf{w}_i\}$ denotes the set of all admissible beamforming vectors. The weighted EE for the considered system is expressed as

$$\mathrm{EE}\left(\{\mathbf{w}_i\}\right) = \frac{\sum_{k=1}^{K} \alpha_k R_k\left(\{\mathbf{w}_i\}\right)}{\sum_{k=1}^{K} \|\mathbf{w}_i\|^2 + P_{\mathrm{C}}}, \tag{3}$$

where $\alpha_k > 0$ is a preassigned weight for the k-th MU and P_{C} denotes the constant power consumption caused by circuit modules.

Our goal is to maximize the EE of the considered network, which is mathematically formulated as an optimization problem:

$$\{\mathbf{w}_i^\star\} = \arg \max_{\{\mathbf{w}_i \in \mathbb{C}^{N_{\mathrm{T}}}\}, \sum_{i=1}^{K} \|\mathbf{w}_i\|^2 \leq P_{\max}} \mathrm{EE}\left(\{\mathbf{w}_i\}\right), \tag{4}$$

where $P_{\max}$ denotes the power budget of the transmitter, and $\|\cdot\|$ denotes the Euclidien norm.

The problem (4) can be efficiently solved by existing CVX techniques but without close-form solution. By treating the CVXopt-based algorithm as a "black box", it maps CSI to beamforming vectors via iterative computations. Such a mapping can also be regarded as a "function" represented by $\Pi(\cdot) : \mathbb{C}^{N_{\mathrm{T}} \times K} \rightarrow \mathbb{C}^{N_{\mathrm{T}} \times K}$. Following the universal approximation theorem, we intend to utilize neural networks to solve the problem (4).

III. Structure of KANsformer

The main idea of the proposed KANsformer is to realize the mapping, i.e., $\Pi(\cdot)$, from $\{\mathbf{h}_i\}$ to $\{\mathbf{w}_i\}$ such that $\mathrm{EE}(\{\mathbf{w}_i\})$ is close to $\mathrm{EE}(\{\mathbf{w}_i^\star\})$. Particularly, we utilize unsupervised learning to train the KANsformer to alleviate the burden on collecting labelled training set. Denote $\boldsymbol{\theta}$ as learnable parameters of the KANsformer, the loss function to update the learnable parameters is given by

$$\mathcal{L}\left(\boldsymbol{\theta}\right) = -\mathrm{EE}\left(\Pi\left(\{\mathbf{h}_i\}\,|\boldsymbol{\theta}\right)\right). \tag{5}$$

Note that via off-line training based on historical statistics, the KANsformer can derive the solution instantaneously at a low computational complexity instead of complex iterative calculation by the mathematical optimization approaches.

The structure of the KANsformer is illustrated in Fig. 1, which includes four modules: pre-processing module, Transformer encoder module, KAN decoder module and post-processing module. The detailed processes in each module are described as follows.

A. Pre-Processing Module

In pre-processing module, we divide each complex-valued CSI vector in $\{\mathbf{h}_k\}$ into its real part, i.e., $\mathrm{Re}(\mathbf{h}_k)$, and its imaginary part, i.e., $\mathrm{Im}(\mathbf{h}_k)$. Then, $\mathrm{Re}(\mathbf{h}_k)$ and $\mathrm{Im}(\mathbf{h}_k)$ are concatenated and input into a linear transformation to obtain the input for the Transformer encoder module expressed as

$$\begin{aligned}\widehat{\mathbf{H}} = &[\mathrm{Con}\left(\mathrm{Re}\left(\mathbf{h}_1\right), \mathrm{Im}\left(\mathbf{h}_1\right)\right); \cdots; \\ &\mathrm{Con}\left(\mathrm{Re}\left(\mathbf{h}_K\right), \mathrm{Im}\left(\mathbf{h}_K\right)\right)] \mathbf{W}_0 \in \mathbb{R}^{K \times D},\end{aligned} \tag{6}$$

where $\mathrm{Con}(\cdot)$ represents the concatenation operation, and $\mathbf{W}_0 \in \mathbb{R}^{2N_{\mathrm{T}} \times D}$ denotes the learnable parameters with D denoting a configurable dimension which is usually greater than K such that more attention heads can be employed.

B. Transformer Encoder Module

The aim of the Transformer encoder module is to encode the obtained network feature $\widehat{\mathbf{H}}$ by exploring interactions among MUs and embedding the impact of inter-MU interference into the encoded network feature. The Transformer encoder module comprises L Transformer encoder layers (TELs), each of which includes two submodules, i.e., multi-head self-attention and position-wise feed-forward. For the l-th TEL, we denote its input and output[2] as $\mathbf{H}^{(l)}$ and $\mathbf{H}^{(l+1)} \in \mathbb{R}^{K \times D}$, respectively. The detailed processes of the two submodules are given as follows.

1) Multi-Head Self-Attention: Suppose that $M^{(l)}$ self-attention heads are employed in the l-th TEL, and then, the attention coefficient matrix associated with the m-th self-attention head in the l-th TEL is given by

$$\begin{aligned}\mathbf{A}_m^{(l)} = &\mathrm{Softmax}\left(\frac{\mathbf{H}^{(l)}\mathbf{W}_Q^{(l)}\left(\mathbf{H}^{(l)}\mathbf{W}_K^{(l)}\right)^T}{\sqrt{D}}\right)\mathbf{H}^{(l)}\mathbf{W}_V^{(l)} \\ &\in \mathbb{R}^{K \times \frac{D}{M^{(l)}}},\end{aligned} \tag{7}$$

where $\mathbf{W}_Q^{(l)}$, $\mathbf{W}_K^{(l)}$ and $\mathbf{W}_V^{(l)} \in \mathbb{R}^{D \times \frac{D}{M^{(l)}}}$ denotes the learnable parameters for the query, key and value projections, respec-

[2]The inputs and outputs of all the TELs have the same size.

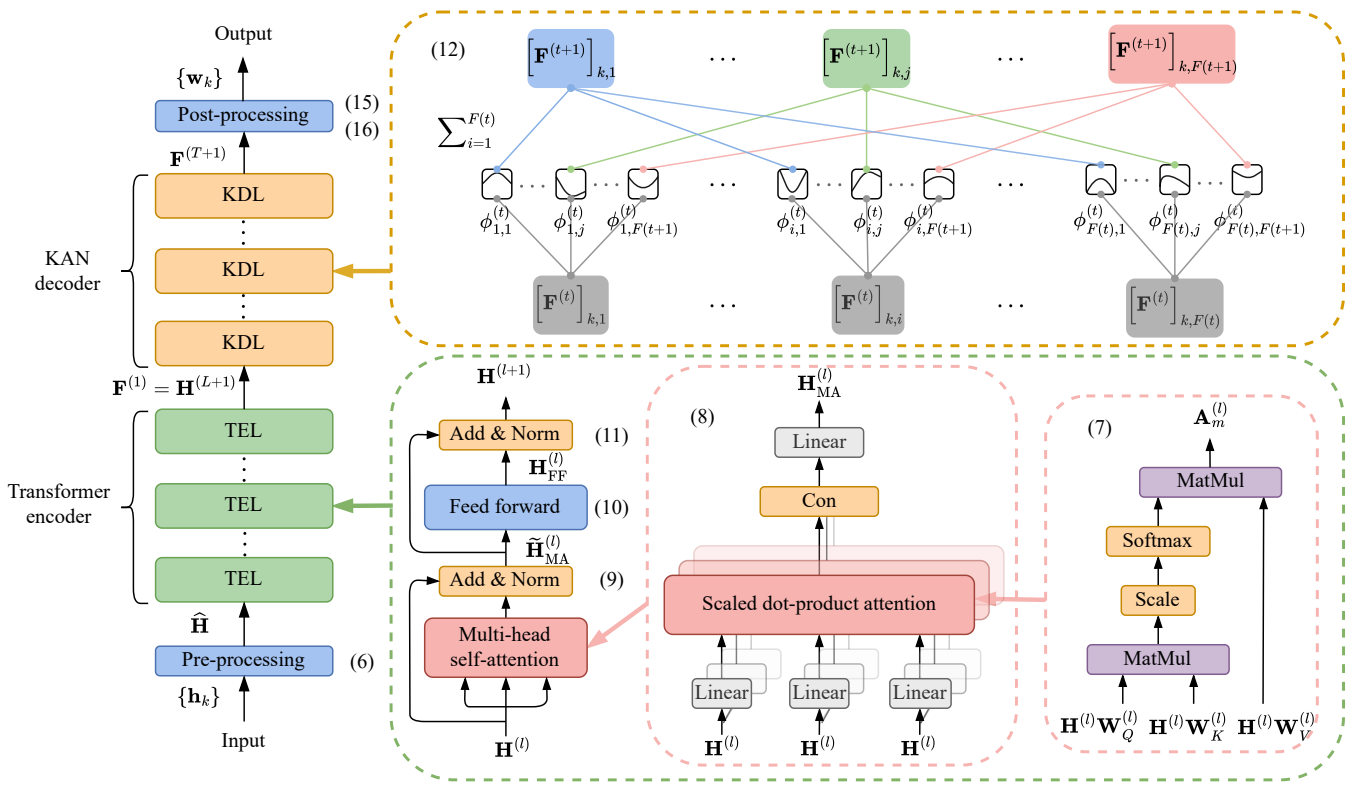

Fig. 1. Structure of KANsformer which includes four modules: pre-processing module, Transformer encoder module (with L TELs), KAN decoder module (with T KDLs) and post-processing module. The detailed processes of the l-th TEL and the t-th KDL are illustrated.

tively. Note that a larger value of D enables the utilization of more attention heads.

The obtained $M^{(l)}$ attention heads, i.e., $\{\mathbf{A}_m^{(l)}\}_m$, are concatenated and then, passed into a linear layer with leanrable parameters of $\mathbf{W}_{\mathrm{MA}}^{(l)} \in \mathbb{R}^{D\times D}$. Then, we obtain the multi-head attention coefficient matrix as

$$\mathbf{H}_{\mathrm{MA}}^{(l)} = \underbrace{\mathrm{Con}\left(\mathbf{A}_1^{(l)} \cdots \mathbf{A}_{M^{(l)}}^{(l)}\right)}_{\in \mathbb{R}^{K\times D}} \mathbf{W}_{\mathrm{MA}}^{(l)} \in \mathbb{R}^{K\times D}. \tag{8}$$

To improve the training performance and stack deeper layers, the parameter-free layer normalization process represented by $\mathrm{LayerNorm}(\cdot)$ and the residual connection are adopted. The attention coefficient matrix is then updated by

$$\widetilde{\mathbf{H}}_{\mathrm{MA}}^{(l)} = \mathrm{LayerNorm}\left(\mathbf{H}_{\mathrm{MA}}^{(l)}\right) + \mathbf{H}^{(l)} \in \mathbb{R}^{K\times D}. \tag{9}$$

2) Position-wise Feed-forward Layer: The obtained attention coefficient matrix is input into a 2-layer feed-forward network with position-wise operation, i.e.,

$$\mathbf{H}_{\mathrm{FF}}^{(l)} = \mathrm{Con}\left(f_2^{(l)}\left(\mathrm{ReLu}\left(f_1^{(l)}\left(\left[\widetilde{\mathbf{H}}_{\mathrm{MA}}^{(l)}\right]_{k,:}\right)\right)\right)\right) \in \mathbb{R}^{K\times D}, \tag{10}$$

where $f_1^{(l)}(\cdot): \mathbb{R}^{D} \to \mathbb{R}^{D'}$ and $f_2^{(l)}(\cdot): \mathbb{R}^{D'} \to \mathbb{R}^{D}$ denote the feed-forward functions with D' denoting an intermediate dimension, dimension, and $[\mathbf{X}]_{k,:}$ denotes the k-th row of $\mathbf{X}$. The learnable parameters of $f_1^{(l)}(\cdot)$ and $f_2^{(l)}(\cdot)$ are denoted as $\mathbf{W}_1^{(l)} \in \mathbb{R}^{D\times D'}$ and $\mathbf{W}_2^{(l)} \in \mathbb{R}^{D'\times D}$, respectively.

Similarly, the layer normalization process and the residual connection are followed by the feed-forward network, and the output of the l-th TEL is given by

$$\mathbf{H}^{(l+1)} = \mathrm{LayerNorm}\left(\mathbf{H}_{\mathrm{FF}}^{(l)}\right) + \mathbf{H}_{\mathrm{FF}}^{(l)} \in \mathbb{R}^{K\times D}. \tag{11}$$

C. KAN Decoder Module

The aim of the KAN decoder module is to decode the obtained network features, i.e., $\mathbf{H}^{(L+1)}$, to the required beamforming vectors via T KAN decoder layers (KDLs). For the t-th KDL, we denote its input and output as $\mathbf{F}^{(t)} \in \mathbb{R}^{K\times F(t)}$ and $\mathbf{F}^{(t+1)} \in \mathbb{R}^{K\times F(t+1)}$, respectively, where $F(t)$ and $F(t+1)$ denote the corresponding dimensions. Note that $\mathbf{F}^{(1)} = \mathbf{H}^{(L+1)}$ and $F(1) = D$ while $F(T+1) = 2N_{\mathrm{T}}$. The processing of the t-th KDL is given by

$$\left[\mathbf{F}^{(t+1)}\right]_{k,j} = \sum\nolimits_{i=1}^{F(t)} \phi_{j,i}^{(t)}\left(\left[\mathbf{F}^{(t)}\right]_{k,i}\right), \tag{12}$$

where $[\mathbf{X}]_{k,j}$ denotes the element in the k-th row and the j-th column of $\mathbf{X}$, $k \in \{1,...,K\}$, $j \in \{1,...,F(t+1)\}$, and $\phi_{j,i}^{(t)}(\cdot): \mathbb{R} \to \mathbb{R}$ is a continuous function which is given by

$$\phi_{j,i}^{(t)}(x) = \beta_{j,i}^{(t)} \frac{x}{1+\exp(-x)} + \gamma_{j,i}^{(t)} \mathrm{Spline}_{j,i}^{(t)}(x), \tag{13}$$

where $\beta_{j,i}^{(t)}$ and $\gamma_{j,i}^{(t)}$ are learnable parameters, and $\mathrm{Spline}_{j,i}^{(t)}(\cdot): \mathbb{R} \to \mathbb{R}$ is parameterized as a linear combination of B-splines such that

$$\mathrm{Spline}_{j,i}^{(t)}(x) = \sum\nolimits_{p=0}^{P} c_{p,j,i}^{(t)} B_p(x), \tag{14}$$

where $c_{p,j,i}^{(t)}$ denotes the learnable weights and P is a hyper-parameter related to the B-splines (cf. [15]).

D. Post-Processing Module

The post-processing module is to convert $\mathbf{F}^{(T+1)}$ obtained by the KAN decoder module into a feasible solution to the problem (4).

In particular, the real-valued $\mathbf{F}^{(T+1)}$ is used to recover K complex-valued beamforming vectors with the k-th beamforming vector given by

$$\widetilde{\mathbf{w}}_k = \mathbf{F}^{(T+1)}\left[k, 1:N_{\mathrm{T}}\right] + i\mathbf{F}^{(T+1)}\left[k, N_{\mathrm{T}}+1:2N_{\mathrm{T}}\right]. \tag{15}$$

Then, each beamforming vector is fed into a scale function to satisfy the power budget of $P_{\max}$:

$$\mathbf{w}_k = \sqrt{\frac{P_{\max}}{\max\left(P_{\max}, \sum_{i=1}^{K}\|\widetilde{\mathbf{w}}_i\|^2\right)}}\widetilde{\mathbf{w}}_k. \tag{16}$$

All the learnable parameters in the KANsformer are collected in the hyperparameter set

$$\boldsymbol{\theta} = \left\{\mathbf{W}_0, \mathbf{W}^{(l)}, \beta_{j,i}^{(t)}, \gamma_{j,i}^{(t)}, c_{p,j,i}^{(t)}\right\}, \tag{17}$$

where $\mathbf{W}^{(l)} \triangleq \{\mathbf{W}_Q^{(l)}, \mathbf{W}_K^{(l)}, \mathbf{W}_V^{(l)}, \mathbf{W}_O^{(l)}, \mathbf{W}_1^{(l)}, \mathbf{W}_2^{(l)}\}$. Note that $\boldsymbol{\theta}$ is independent of K, thus facilitating the KANsformer to accept the input $\{\mathbf{h}_k\}$ with different values of K.

IV. Numerical Results

This section provides numerical results to evaluate the proposed KANsformer in terms of generalization performance, transfer learning and ablation experiment under the following settings.

1) Simulation scenario: All the system parameters used are $N_{\mathrm{T}} \in \{4, 8, 16\}$, $K \in \{2, 3, 4, 5, 7, 8, 9, 10, 12, 14\}$, $\alpha_k = 1$ ($\forall k \in \mathcal{K}$), $P_{\max} = 1$ W, $P_{\mathrm{C}} = 0.1$ W, CSI $\{\mathbf{h}_k \in \mathbb{C}^{N_{\mathrm{T}}}\}$ being Rayleigh distributed for both training samples and test samples, and the corresponding labels (for test samples) representing the maximal EEs obtained by CVXopt-based algorithms. Specifically, we use K_{Tr}, K_{Te} and K'_{Tr} to respectively denote the number of MUs in the training stage, test stage and fine-tuning training stage (due to transfer learning), where $K_{\mathrm{Te}} \neq K_{\mathrm{Tr}}$ (known as scalability) is unknown during the training stage.

2) Structure of KANsformer: The KANsformer under test comprises $L = 4$ TELs and $T = 4$ KDLs. Each TEL has the dimension of input/output features as $D = 8 \times N_{\mathrm{T}}$, the number of attention heads as $M^{(l)} = 4$, and the intermediate dimension of the feed-forward layer as $D' = 32 \times N_{\mathrm{T}}$. The input dimensions of KDLs are given by $F(1) = 8 \times N_{\mathrm{T}}$, $F(2) = 256$, $F(3) = 128$ and $F(4) = 64$, respectively, while the output dimension of the last KDL is given by $F(5) = 2 \times N_{\mathrm{T}}$.

3) Computer configuration: All DL models are trained and tested by Python 3.10 with Pytorch 2.4.0 on a computer with Intel(R) Xeon(R) Platinum 8255C CPU and NVIDIA RTX 2080 Ti (11 GB of memory).

TABLE I
Generalization performance evaluation.

N_{T}	K_{Tr}	K_{Te}	CVX	MLP	GAT	KF†
4	2	2	100%	98.2%	98.4%	**99.5%**
Inference time			6.7s	**3.3 ms**	7.7 ms	8.5 ms
8	4	3	×	×	84.3%	**85.1%**
		4	100%	79.1%	90.1%	**95.3%**
		5	×	×	82.6%	**83.2%**
Inference time			10.5 s	**3.4 ms**	7.7 ms	8.5 ms
16	8	7	×	×	84.0%	**91.1%**
		8	100%	17.9%	85.6%	**92.9%**
		9	×	×	82.8%	**90.8%**
Inference time			57.4 s	**3.5 ms**	7.8 ms	8.4 ms

†KF is short for KANsformer.
× represents "not applicable".

4) Initialization and training: The learnable parameters are initialized according to He (Kaiming) method and the learning rate is initialized as 10^{-4}. The $Adam$ algorithm is adopted as the optimizer during the training phase. The batch size is set to 16 for 100 training epochs. The learnable weights with the best performance are used as the training results.

5) Benchmark DL models: In order to evaluate the KANsformer numerically, the following four baselines are considered, i.e.,

- CVXopt-based approach: A single-layer successive convex approximation based optimization algorithm, similar to Algorithm 1 in [16], used to generate the test labels.
- MLP: A basic feed-forward neural network, similar to [2].
- GAT[3]: A basic GCN with multi-head attention mechanism, similar to [6].

6) Test performance metrics:

- Optimality performance: The ratio of the average achievable EE by the DL model to the optimal EE.
- Inference time: Average running time for yielding the feasible beamforming solution by the DL model.

A. Generalization Performance

The numerical performance tests and inference times of the KANsformer are given in Table I, which are presented in more detail below, respectively.

1) Optimality performance with $K_{\mathrm{Te}} = K_{\mathrm{Tr}}$ (marked by blue-shaded areas): One can observe that the KANsformer outperforms the MLP and the GAT for all the three cases; the larger of N_{T} and $K_{\mathrm{Te}} = K_{\mathrm{Tr}}$, the larger the performance degradation, showing larger negative impact on the learning performance for all the DL models. However, the KANsformer with the best performance maintains the performance loss within 10%.

[3]As demonstrated in [6] and [17], the GAT generally outperforms the CNN and the GCN. Therefore, we select the GAT as a baseline.

TABLE II
TRANSFER LEARNING EVALUATION: $N_{\mathrm{T}} = 16$.

K_{Te}	Scaling† ($K_{\mathrm{Tr}} = 8$)	Re-training ($K_{\mathrm{Tr}} = K_{\mathrm{Te}}$)	Transfer learning ($K_{\mathrm{Tr}} = 8$, $K'_{\mathrm{Tr}} = K_{\mathrm{Te}}$)		
		100 epochs	10 epochs	20 epochs	50 epochs
10	86.6%	93.4%	93.0%	**93.5%**	**93.5%**
12	77.9%	90.7%	94.6%	**95.2%**	**95.2%**
14	71.1%	**95.2%**	93.6%	94.2%	94.2%

†Scaling represents that directly applying the model trained with K_{Tr} to the scenario of K_{Te}.

TABLE III
ABLATION EXPERIMENT: $N_{\mathrm{T}} = 16$ AND $K_{\mathrm{Tr}} = 8$.

Encoder		Decoder		K_{Te}		
GAT	TF†	MLP	KAN	7	8	9
✓	×	✓	×	84.0%	85.6%	82.8%
✓	×	×	✓	89.5%	91.2%	88.1%
×	✓	✓	×	82.5%	85.8%	83.2%
×	✓	×	✓	**91.1%**	**92.9%**	**90.8%**
Avg. gain		-		0.1%	1.9%	3.1%
-		Avg. gain		14.1%	7.3%	12.9%

†TF is short for Transformer.

2) Optimality performance with $K_{\mathrm{Te}} \neq K_{\mathrm{Tr}}$ (marked by orange-shaded areas): The KANsformer performs much better than the GAT for $K_{\mathrm{Tr}} = 8, /K_{\mathrm{Te}} \in \{7, 9\}$, but slightly better for $K_{\mathrm{Tr}} = 4, /K_{\mathrm{Te}} \in \{3, 5\}$, besides some performance loss compared with the case for $K_{\mathrm{Te}} = K_{\mathrm{Tr}} \in \{4, 8\}$. These results also indicate that the scalability performance loss is larger for larger $|K_{\mathrm{Te}} - K_{\mathrm{Tr}}|/K_{\mathrm{Tr}}$, because the multi-head self-attention mechanism intends to explore the interaction among MUs, which may change with the number of MUs.

3) Inference time: All of the MLP, GAT and KANsformer achieve millisecond-level inference (significantly faster than the iterative CVXopt-based approach) such that they are applicable under time-varying channel conditions. A more surprising observation is that the inference time of DL models remains almost unchanged for all the numbers of N_{T} and K used, while it increases exponentially for the CVXopt-based approach (a widely known fact).

In summary, the well-trained KANsformer can achieve real-time and near-optimal inference for solving the problem (4), while being scalable to the number of MUs (though K_{Te} unknown in the training stage) and remaining acceptable performance in the meantime.

B. Transfer Learning

As mentioned that the scalability suffers from performance degradation with the increment of $|K_{\mathrm{Te}} - K_{\mathrm{Tr}}|/K_{\mathrm{Tr}}$. One can retrain the model or fine-tune the model via transfer learning on a new dataset (where the number of users is K_{Te}). The former initializes the learnable parameters randomly while the latter adopts the learnable parameters of the model trained for K_{Tr} as the initial values of the model $\boldsymbol{\theta}$ instead. Table II shows the performance of scaling, re-training (via 100 epochs) and transfer learning (via $\{10, 20, 50\}$ epochs). It can be seen that the transfer learning can effectively improve the performance at a quite low training cost (e.g., 10 epochs) compared with the performance of the plain scaling, meanwhile achieving a comparable performance of the re-training at fewer training epochs (e.g., 20 epochs). For $K_{\mathrm{Te}} = 14$, the transfer learning falls behind the re-training by 1%, and the reason is that the prior-knowledge for K_{Tr} may mislead the transfer learning under $K'_{\mathrm{Te}} = K_{\mathrm{Tr}}$ with large $|K'_{\mathrm{Tr}} - K_{\mathrm{Tr}}|$. Nevertheless, the transfer learning can also achieve a considerable performance gain ($> 20\%$) compared with the plain scaling.

C. Ablation Experiment

Table III gives the ablation experiment to validate the effectiveness of Transformer used as the encoder and KAN used as the decoder. A performance gain can be observed by comparing Transformer/KAN and GAT/MLP for both cases of $K_{\mathrm{Te}} = K_{\mathrm{Tr}}$ and $K_{\mathrm{Te}} \neq K_{\mathrm{Tr}}$. Specifically, the average performance gains over $K_{\mathrm{Te}} \in \{7, 8, 9\}$ resulting from Transformer and KAN are respectively 1.7% and 11.4%. The reason for this is that both the GAT and Transformer adopt the attention mechanism to enhance the expressive capability while KAN has more flexible activation processes than MLP, such that KAN can outperform MLP in terms of interpretability [13].

V. CONCLUSION

We have presented a DL model (i.e., the KANsformer shown in Fig. 1) with Transformer and KAN used in the encoder-decoder structure, respectively, for solving the beamforming design problem (cf. (4)).

Numerical results showed that the KANsformer outperforms the existing DL models in terms of both the performance accuracy and the inference time consumed. Furthermore, we would like to emphasize that, in response to the given input CSI $\{\mathbf{h}_k\}$, the KANsformer can yield the beamforming vector output $\{\mathbf{w}_k\}$, with the elapsed inference time almost fixed (thus insensitive to the problem size) and tremendously lower than the problem-size dependent running time required by the CVXopt-based approach; the performance accuracy of the former is quite close to that the latter (treated as the optimum). These results also motivate worthy further development of more powerful encoders and decoders dedicated to wireless communication systems.

REFERENCES

[1] Y. Lu, W. Mao, H. Du, O. A. Dobre, D. Niyato, and Z. Ding, "Semantic-aware vision-assisted integrated sensing and communication: Architecture and resource allocation," *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 302-308, Jun. 2024.

[2] C. Hu et al., "AI-empowered RIS-assisted networks: CV-enabled RIS selection and DNN-enabled transmission," *IEEE Trans. Veh. Technol*, vol. 73, no. 11, pp. 17854-17858, Nov. 2024.
[3] Z. Song et al., "A deep learning framework for physical-layer secure beamforming," *IEEE Trans. Veh. Technol*, vol. 73, no. 12, pp. 19844-19849, Dec. 2024.
[4] Y. Lu, Y. Li, R. Zhang, W. Chen, B. Ai, and D. Niyato, "Graph neural networks for wireless networks: Graph representation, architecture and evaluation," *IEEE Wireless Commun.*, vol. 32, no. 1, pp. 150-156, Feb. 2025.
[5] Y. Shen, J. Zhang, S. H. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 3554-3569, May 2023.
[6] Y. Li, Y. Lu, R. Zhang, B. Ai, and Z. Zhong, "Deep learning for energy efficient beamforming in MU-MISO networks: A GAT-based approach," *IEEE Wireless Commun. Lett.*, vol. 12, no. 7, pp. 1264-1268, July 2023.
[7] Y. Li, Y. Lu, B. Ai, O. A. Dobre, Z. Ding, and D. Niyato, "GNN-based beamforming for sum-rate maximization in MU-MISO networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 9251-9264, Aug. 2024.
[8] Y. Li, Y. Lu, B. Ai, Z. Zhong, D. Niyato, and Z. Ding, "GNN-enabled max-min fair beamforming," *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 12184-12188, Aug. 2024.
[9] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 101-115, Jan. 2021.
[10] C. He, Y. Li, Y. Lu, B. Ai, Z. Ding, and D. Niyato, "ICNet: GNN-enabled beamforming for MISO interference channels with statistical CSI," *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 12225-12230, Aug. 2024.
[11] A. Vaswani, Ashish et al., "Attention is all you need," in Proc. *Neurlps*, pp. 5998-6008, 2017.
[12] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and Z. Gao, "UAV trajectory planning for AoI-minimal data collection in UAV-aided IoT networks by Transformer," *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 1343-1358, Feb. 2023.
[13] Z. Liu, Y. Wang, S. Vaidya, et al., "Kan: Kolmogorov-Arnold networks," *arXiv preprint*: 2404.19756, Apr. 2024.
[14] Y. Lu, K. Xiong, P. Fan, Z. Ding, Z. Zhong, and K. B. Letaief, "Global energy efficiency in secure MISO SWIPT systems with non-linear power-splitting EH model," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 1, pp. 216-232, Jan. 2019.
[15] L. Schumaker, "Spline functions: Basic theory." Wiley, 1981.
[16] Y. Lu, "Secrecy energy efficiency in RIS-assisted networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 12419-12424, Sept. 2023.
[17] R. Zhang, Y. Lu, W. Chen, B. Ai, and Z. Ding, "Model-based GNN enabled energy-efficient beamforming for ultra-dense wireless networks," early accessed in *IEEE Trans. Wireless Commun.*, 2025.