Transformer-Empowered 6G Intelligent Networks: From Massive MIMO Processing to Semantic Communication

Yang Wang, Zhen Gao, Dezhi Zheng, Sheng Chen, *Fellow, IEEE*,
Deniz Gündüz, *Fellow, IEEE*, and H. Vincent Poor, *Life Fellow, IEEE*

***Abstract*—6G wireless networks are foreseen to speed up the convergence of the physical and cyber worlds and to enable a paradigm-shift in the way we deploy and exploit communication networks. Machine learning, in particular deep learning (DL), is going to be one of the key technological enablers of 6G by offering a new paradigm for the design and optimization of networks with a high level of intelligence. In this article, we introduce an emerging DL architecture, known as the *transformer*, and discuss its potential impact on 6G network design. We first discuss the differences between the transformer and classical DL architectures, and emphasize the transformer's self-attention mechanism and strong representation capabilities, which make it particularly appealing in tackling various challenges in wireless network design. Specifically, we propose transformer-based solutions for various massive multiple-input multiple-output (MIMO) and semantic communication problems, and show their superiority compared to other architectures. Finally, we discuss key challenges and open issues in transformer-based solutions, and identify future research directions for their deployment in intelligent 6G networks.**

I. Introduction

The sixth generation (6G) of wireless cellular networks are expected to connect the cyber and physical worlds, allowing humans to seamlessly interact with a variety of devices in a mixed reality metaverse through connected intelligence. These new and fascinating applications impose challenging requirements and constraints on communication networks, including ultra-high reliability, ultra-low latency, extremely high data rate, substantially high energy and spectral efficiency, ultra-dense connectivity, and a high level of intelligence. These stringent demands of 6G have driven researchers to look for sophisticated physical layer techniques that would go beyond the cycle of incremental improvements. Current wireless networks have been largely designed as a combination of dedicated processing blocks, such as channel estimation, equalization, coding/decoding blocks, where each block is designed separately on the basis of mathematical models that define the statistical behavior of the wireless channels and the underlying data traffic. However, this model-driven and block-based design approach is facing increasing challenges in the complex and diversified scenarios the future 6G networks will operate in. The diversity of the devices and hardware technologies, increasing co-existence requirements, and the variety of traffic and service demands make such modeling approach difficult and inaccurate. In addition, with the deployment of ultra-massive multiple-input multiple-output (MIMO) systems and large-scale reconfigurable intelligent surfaces (RISs), the optimization of physical layer functionalities based on rigid mathematical models and solutions become prohibitive due to the computational complexity and associated control overheads. Therefore, conventional mathematical models and solutions cannot provide the required dramatic enhancement in the capacity and performance of wireless networks.

Recently, machine learning, in particular deep learning (DL), has emerged as a powerful alternative for the design and optimization of wireless networks by learning the underlying statistical structures from data instead of building and employing accurate mathematical models [1]. The potential impact of DL-based solutions have already been shown in a variety of challenging wireless communication problems, in which it is either difficult to obtain a model of the system, or the complexity of the model does not lend itself to tractable solutions with feasible computational complexity [1], [2].

While DL-based solutions are appealing, the actual deployment is still challenging as they require architecture and hyperparameter optimization for each specific task. Therefore, proposing a more efficient and widely applicable DL architecture is essential for solving complex communication problems. A novel deep neural network (DNN) structure, called the *transformer*, has emerged recently, and achieved remarkable success in a variety of natural language processing (NLP) and computer vision (CV) tasks [4]. The transformer architecture is built upon the *self-attention* mechanism, which relates different parts of a data sequence for a more accurate representation of the sequence. Self-attention layers in the transformer architecture enable a global receptive field, and the multi-head mechanism ensures that the network can pay attention to multiple discriminative parts of the inputs. By highlighting the transformer's multi-model fusion and feature representation capabilities, we explore its application in 6G intelligent network design, and propose a new transformer-based intelligent processing architecture. We focus on massive MIMO and semantic communication applications; however, we expect the transformer architecture to find applications in many other components of future data-driven 6G networks.

The rest of the article is organized as follows. The following section briefly introduces the application of DL in wireless communications. Then, we introduce the transformer architecture. Next, we present a transformer-based architecture for 6G intelligent processing, and study its performance in various wireless communication problems. We then discuss

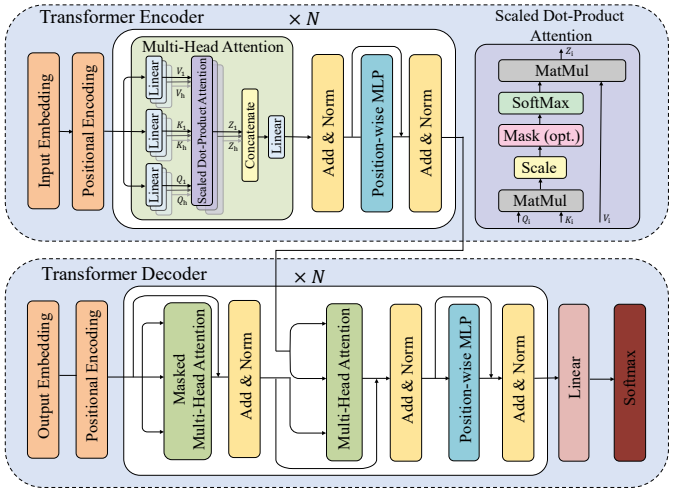


Fig. 1. Structure of the transformer network.

open research issues in transformer-empowered 6G intelligent networks and conclude the article.

II. Overview of Deep Learning and the Transformer Architecture

DL is a powerful computational tool to understand complex data representations and patterns, and as such, offers a new paradigm to tackle complicated problems in 6G intelligent network design. In this section, we briefly provide some background on popular DNN architectures and their applications in wireless communications.

A. Common DNN Architectures

Classic neural network architectures include multi-layer perception (MLP), convolutional neural network (CNN), recurrent neural network (RNN), and stacked autoencoder (SAE).

MLP is an artificial neural network that consists of at least three layers of fully-connected neurons, parameterized by a substantial number of connection weights. Under the premise of keeping the same input and output dimensions, the computational complexity of the fully-connected layer is given by $\mathcal{O}(n^2 \cdot d^2)$[1], where the input vector $\mathbf{x} \in \mathbb{R}^{1 \times nd}$ is reshaped from the two-dimensional sequence $\mathbf{X} \in \mathbb{R}^{n \times d}$. MLP-based solutions have been developed to address various wireless communication problems, such as channel estimation and beamforming [2]. It has been observed that deeper networks typically provide better generalization; however, training fully-connected deep networks suffers from high complexity and low convergence efficiency.

To reduce the training complexity, CNNs employ a set of locally connected kernels, rather than fully-connected layers, to capture local correlations between different data regions. Compared with MLP, CNN reduces the number of model parameters significantly and maintains the affine invariance by leveraging three important ideas: sparse interactions, parameter sharing, and equivariant representations. The computational complexity for the convolutional layer is given by $\mathcal{O}(k \cdot n \cdot d^2)$, where $k \times d$ is the kernel size to adapt sequential processing [4]. By treating the channel matrices as two-dimensional images, CNNs have shown great potential for tasks such as channel estimation, channel state information (CSI) feedback, beamforming [2], as well as semantic image transmission [3].

RNNs constitute another class of DNN architectures that exploit sequential correlations between samples. At each step, it produces the output via recurrent connections between hidden units. However, the traditional RNN architecture is slow to train, and suffers from vanishing and exploding gradients. Long short-term memory (LSTM) architecture mitigates these problems by introducing a set of gates, which allow memory to be restored across longer sequences. The computational complexity for the recurrent layer is given by $\mathcal{O}(n \cdot d^2)$ [4]. Recently, there have been several works utilizing LSTMs to extract temporal correlations across data, (e.g., in channels with memory) for communication system design [2].

SAE architecture consists of hierarchically connected multiple autoencoders. Its basic component, autoencoder, contains two parts: an encoder that acquires a low-dimensional representation of input, and a decoder that reconstructs the input from the compressed vector. SAE is widely used to extract features and patterns that contain essential and compressed information about data. From a learning perspective, the entire communication system can be viewed as an end-to-end SAE, and its multiple sub-modules can also be viewed as SAEs, including pilot design and channel estimation, CSI feedback, and semantic communications [2], [3]. Thus, SAE is a core DNN structure for many of the current DL-based communication system components.

B. Self-Attention and Transformer

Although MLP, CNN, RNN, and SAE have been widely utilized in DL-based communication system design with some success, efforts continue to push the boundaries of DL models in practical communication systems. Recently, the evolution of DNN architectures in NLP has led to a prevalent architecture known as the transformer [4]. We argue that the transformer holds a great potential also in the design of intelligent communication systems.

As shown in Fig. 1, the transformer is a sequence-to-sequence DNN model and consists of an encoder and a decoder module with several encoder/decoder layers of the same architecture. The input and output sequences are converted to vectors of dimension d by embedding and positional encoding layers. Each encoder/decoder layer has the same structure, and is mainly composed of a *self-attention* sub-layer following by a position-wise MLP sub-layer, while each decoder also contains a masked attention sub-layer before the self-attention sub-layer. For building a deep model, a residual connection is employed around each sub-layer, followed by a layer normalization module.

Self-attention mechanism relates different positions in a single sequence to compute a representation of the sequence, which can also be regarded as a non-local filtering operation. In a single-head self-attention layer, the input sequence $\mathbf{X} \in \mathbb{R}^{n \times d}$ is first transformed into three different sequential

[1]Note that, a two-dimensional sequence $\mathbf{X} \in \mathbb{R}^{n \times d}$ is used to analyze the complexity of different DNN structures, where n is the sequence length, and d is the representation dimension.

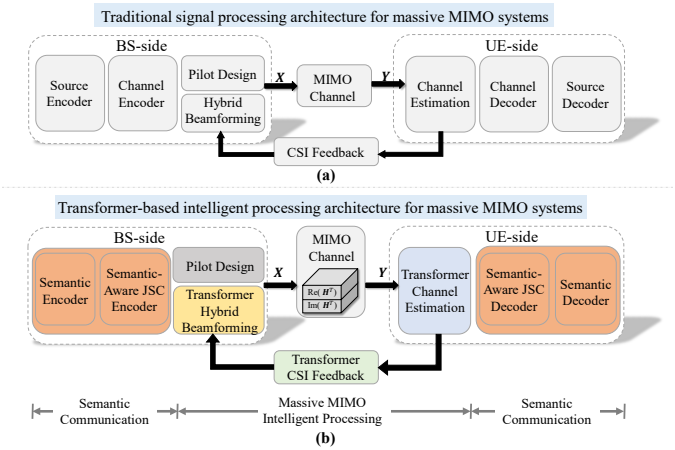


Fig. 2. Traditional and proposed transformer-based signal processing architecture for massive MIMO systems.

vectors: the query $\mathbf{Q} \in \mathbb{R}^{n \times d_k}$, the key $\mathbf{K} \in \mathbb{R}^{n \times d_k}$ and the value $\mathbf{V} \in \mathbb{R}^{n \times d_v}$ by three different linear matrices, which are obtained through training. Here, d_k and d_v are the dimensions of query (key), and value subspaces, respectively. Subsequently, as shown in Fig. 1, the scale dot-production attention operation generates the attention weights by aggregating the query and the corresponding key. The resulting weights are assigned to the corresponding value, yielding the output vectors. To facilitate the complexity analysis, we assume that query, key, and value matrices have the same dimension as the input sequence, i.e., $d_k = d_v = d$. Thus, the complexity of self-attention layer can be expressed as $\mathcal{O}(n^2 \cdot d)$ [4]. In terms of computational complexity, self-attention layers are significantly faster than fully-connected layers, and are faster than recurrent layers when the sequence length n is smaller than the representation dimensionality d. The training efficiency of recurrent layers is much lower than that of the self-attention layers due to the sequential processing. Furthermore, since convolutional layers are generally more complex than recurrent layers, by a factor of k, their complexity is also higher than the self-attentive layer. Instead of performing single-head self-attention with query, key, and value, multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. Specifically, different heads use different three group linear matrices, and these matrices can project the input vectors into multiple feature subspaces (i.e., $\{\mathbf{Q}_i\}_{i=1}^h$, $\{\mathbf{K}_i\}_{i=1}^h$, and $\{\mathbf{V}_i\}_{i=1}^h$, where h is the number of heads) and processes them by several parallel attention heads (layers). The resulting vectors are concatenated and mapped to the final output.

The position-wise MLP sub-layer is a fully-connected feed-forward module that operates separately and identically on each position. This module consists of two linear transformations with ReLU activation, where the parameters are shared across different positions, and the complexity is $\mathcal{O}(n \cdot d^2)$ [4]. Since the transformer does not introduce recurrence or convolution, it has no knowledge of positional information (especially for the encoder). Thus, additional positional information is introduced through positional encoding in order to model the relative positions of the input sequences.

Compared with CNN/RNN models, the transformer makes few assumptions about the underlying structure of data, which makes it a universal and flexible architecture. The non-sequential nature of the transformer architecture allows it to capture long-range dependencies in the input data through self-attention. Not surprisingly transformers have also shown remarkable success in semantic communications [5], which considering the transmission of text source over noisy channels. In this article, we show that transformers can have a critical role in other communication tasks as well.

III. Transformer For 6G Intelligent Processing

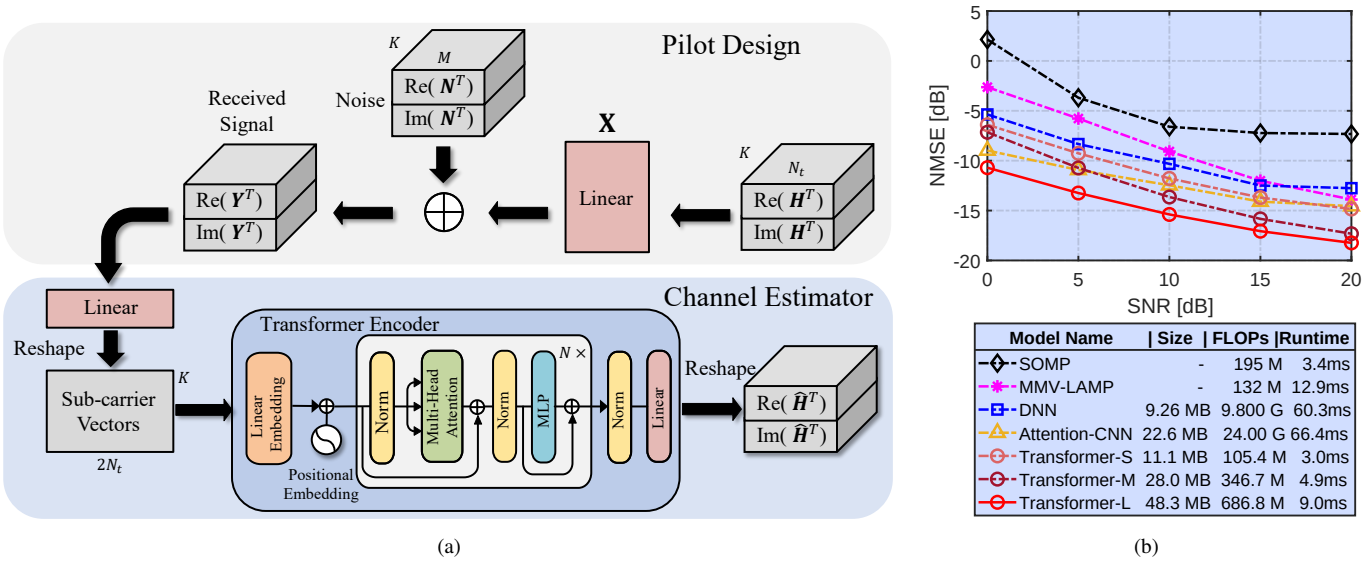
Massive MIMO is an essential physical layer technology to accommodate the exponential growth of mobile data traffic. Fig. 2 (a) illustrates a generic communication system, divided into two parts: the MIMO processing part and source & channel coding part. The former includes pilot design, channel estimation, CSI feedback, and hybrid beamforming (HBF). The latter is composed of source coding and channel coding. We seek to expand the applicability of the transformer to serve as a general-purpose backbone for these crucial modules. In particular, as illustrated in Fig. 2 (b), we propose a novel 6G intelligent processing architecture employing transformer for both the massive MIMO intelligent processing blocks and the newly emerging semantic communication blocks.

A. Channel Estimation

Accurate CSI at the base station (BS) is critical for beamforming and signal detection in massive MIMO systems. However, CSI acquisition overhead of conventional orthogonal pilot approaches increases linearly with the number of antennas. To reduce the pilot overhead, existing 5G NR standard limits the number of pilot signals to be significantly smaller than the number of antennas. However, it is challenging to accurately estimate the high-dimensional channels with low training overhead. By exploiting the sparsity of the channels in the angular domain and/or delay domain, compressive sensing (CS)-based channel estimation solutions have been proposed to overcome this issue. Nevertheless, since the dimension of the CSI to be estimated is extremely large, the involved matrix inversion operations and the iterative nature of CS-based techniques result in prohibitively high computational complexity and storage requirements.

More recently, researchers have resorted to DL techniques to solve the aforementioned problems. A multiple-measurement-vector learned approximate message passing (MMV-LAMP) network was proposed in [6] to reconstruct the spatial-frequency channel matrix by exploiting the channel's structured sparsity. The authors of [7] proposed an end-to-end DNN architecture to jointly design the pilot signals and channel estimator. Moreover, a CNN module combined with non-local attention layer is employed in [8] to exploit longer range correlations in the channel matrix.

Nevertheless, most existing DL-based channel estimation solutions are based on the MLP and CNN architectures. Here, we propose a novel channel estimator that utilizes the universal and flexible transformer architecture, as illustrated in Fig. 3 (a). Specifically, the proposed transformer-based solution includes



(a) (b)

Fig. 3. (a) The transformer-based end-to-end architecture for jointly designing the pilot signals and channel estimator; and (b) NMSE performance comparison of different channel estimation schemes versus SNR.

a dimensionality reduction network for pilot design and a reconstruction network for channel estimation. We exploit a fully-connected linear layer to learn the pilot sequences, which has been widely utilized in the literature [6]–[8]. More importantly, in our channel estimation module, the encoder part of the transformer is exploited to reconstruct the channel. Unlike local-attention in [8], self-attention in the transformer can extract the correlation between subcarriers globally and adjust the weight of each subcarrier, so that the global features of the channel matrix can be extracted for enhanced estimation accuracy.

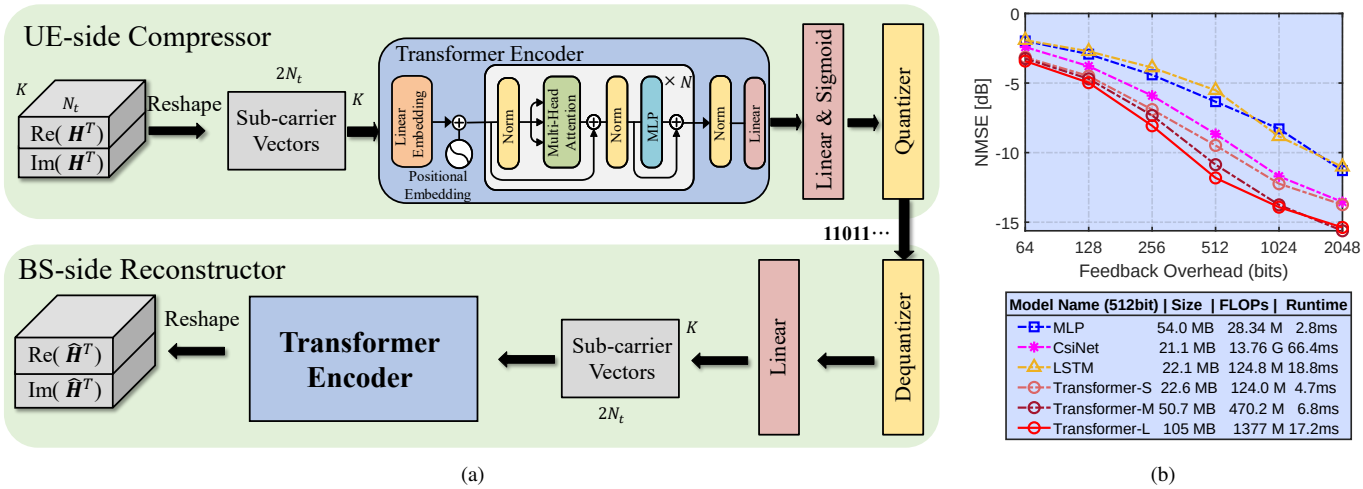
To evaluate the performance of the proposed transformer-based channel estimator, we investigate the downlink channel estimation problem in M successive time slots, where the BS is equipped with a uniform planar array (UPA) with $N_t = 8 \times 8 = 64$ antennas, the user equipment (UE) has single-antenna, the number of orthogonal frequency division multiplexing (OFDM) sub-carriers is $K = 32$, and the channel estimation compression ratio is $\rho = \frac{M}{N_t} = \frac{3}{8}$. We consider a sparse channel scenario with $N_c = 6$ clusters, $N_p = 10$ paths per cluster, and an angle spread of $\Delta\theta = \pm 3.75°$. We generate the training, validation, and test datasets of 100,000, 10,000, 5,000 samples, respectively. We choose the normalized mean square error (NMSE) as the performance metric.

To illustrate the advantages of our proposed channel estimator in Fig. 3 (a), we compare it with four benchmarks. The first one is the traditional simultaneous orthogonal matching pursuit (SOMP) based estimator, denoted as 'SOMP'. The second and third are the conventional DL-based channel estimators, namely, the MMV-LAMP based estimator [6] and the DNN-based estimator [7], denoted as 'MMV-LAMP' and 'DNN', respectively. Finally, we consider the state-of-the-art attention-CNN based channel estimator [8], abbreviated as 'Attention-CNN', as the fourth benchmark. We propose three distinct transformer-based estimators with different model sizes, denoted as 'Transformer-S', 'Transformer-M', and 'Transformer-L', respectively. Fig. 3 (b) shows the NMSE performance of different channel estimation schemes. Evidently, the proposed transformer-based estimator significantly outperforms the conventional and other DL-based methods, especially with comparable model sizes. We observe that, during the inference stage, the floating-point operations per second (FLOPs) and the runtime per sample of the transformer-based estimator are much lower than those of other DL-based methods. Moreover, we can observe that the performance of the transformer improves with the model size. This demonstrates that the transformer-based method can learn latent features from the data more effectively to achieve better channel estimation accuracy with less pilot overhead. It also provides a flexible trade-off between the model complexity and performance, and the users can choose the operating point based on the underlying resources and application requirements.

B. CSI Feedback

CSI feedback is essential in frequency-division duplex (FDD) systems. For time-division duplex (TDD) systems, by exploiting channel reciprocity, the transmitter may estimate the downlink CSI from the uplink CSI. But such reciprocity relies on many ideal factors, including the accurate calibration of the transceiver RF chains at both the BS and UE. For massive MIMO, the perfect uplink and downlink reciprocity is difficult to achieve, and the BS has to rely on CSI feedback for both FDD and TDD operations. However, the large number of antennas result in excessive feedback overhead. Similarly to channel estimation, CS-based techniques can be used to reduce the CSI feedback overhead. However, these techniques cannot fully exploit the channel structure since the channels in real systems are not exactly sparse.

Recently, DL-based solutions have achieved good results in CSI feedback. In [9], bit-level CsiNet, an autoencoder architecture, was proposed to reduce feedback overhead in massive MIMO systems. It has been shown that CsiNet remarkably outperforms traditional CS-based methods in terms of both



(a) (b)

Fig. 4. (a) The transformer-based CSI feedback architecture; and (b) NMSE performance comparison of different CSI feedback schemes versus feedback overhead.

compression ratio and recovery accuracy [9]. Subsequent studies expanded and designed various network models based on CNN and LSTM architectures to handle different CSI feedback problems [2], [10].

Herein, we present a transformer-based CSI feedback scheme to obtain more efficient quantization and compression performance compared with the aforementioned methods. As illustrated in Fig. 4 (a), we utilize the fully-connected linear layer to linearly embed the channel data and use sine and cosine functions of different frequencies to represent the relative positions of the sub-carriers. Then, the transformer encoder extracts the features from the channel data embedded with the positional information. Next, the features are vectorized, and a fully-connected linear layer is used to generate a real-valued compressed codeword. The codeword is then converted to the feedback bit-stream through a quantization layer, which is constructed by uniform scalar quantization [9]. Since the whole network structure corresponds to the compression recovery task, the decoder adopts the same structure as the encoder.

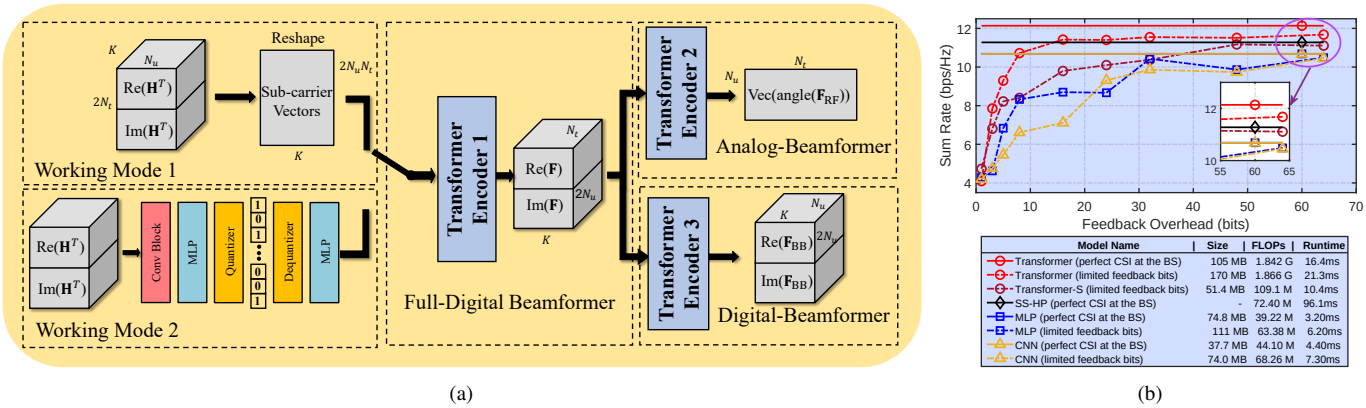
We use the same simulation parameters of Subsection III-A to evaluate the proposed transformer-based CSI feedback schemes with different model sizes, denoted as 'Transformer-S', 'Transformer-M', and 'Transformer-L', respectively. Three benchmark schemes are also considered for comparison. The first one is the MLP-based CSI feedback scheme, 'MLP', where the encoder and decoder consist of three fully-connected layers, respectively. The second one is the 'CsiNet' scheme in [9], while the third is the 'LSTM' scheme in [10]. We again use the NMSE metric for performance evaluation. Fig. 4 (b) shows that all three transformer-based CSI feedback schemes outperform the three benchmarks. Meanwhile, the FLOPs of the transformer-based schemes are much lower than 'CsiNet', and all the proposed schemes have lower runtime than both the 'CsiNet' and the 'LSTM' schemes. Also, we can observe that the performance of the transformer improves with the model size, providing a trade-off between complexity and performance. We can see that 'Transformer-S' is sufficient when a few feedback overhead is desired, while the more complex alternatives provide further gains as the feedback overhead increases. In a nutshell, the transformer can better extract these implicit features in the CSI and fewer feedback bits are needed to reconstruct the CSI at the BS with the same quality, which reduces the feedback overhead and latency.

C. Hybrid Beamforming

Conventional massive MIMO with fully-digital architecture requires a dedicated RF chain for each antenna, which imposes excessive power consumption and extremely high RF hardware cost. The alternative hybrid analog-digital MIMO system employs a much lower number of digital RF chains than the total number of antennas, where each RF chain is connected to multiple active antennas, and the signal phase on each antenna is controlled via a network of analog phase shifters. The analog phase shifter can be seen as a low-cost passive device, which can only control the phase of the signal. Compared with its fully-digital counterpart, HBF optimization is significantly more challenging due to the constant modulus constraint on the analog beamformer [11].

Many model-based solutions have been proposed to tackle this challenge. For instance, the authors of [11] proposed spatial sparse hybrid precoding (SS-HP) to achieve near fully-digital performance by exploiting channel sparsity. However, model-based HBF algorithms require time consuming optimization iterations to obtain near-optimal solutions. Moreover, they demand either perfect downlink CSI or a codebook with an accurate sparse basis, which are difficult to acquire in practice. To overcome these issues, DL-inspired beamforming has been proposed, whereby the prior information is captured from the radio channel measurements. In [12], the authors proposed a CNN-based HBF architecture that can be trained to maximize the spectral efficiency with imperfect CSI. The authors of [13] proposed a MLP-based downlink multi-user HBF module to maximize the spectral efficiency from the limited CSI feedback bits.

To the best of our knowledge, all the existing DL-based HBF schemes adopt the MLP or CNN architectures, and there is still a large gap between their performance and the optimal one. We propose a transformer-based HBF scheme, composed



(a) (b)

Fig. 5. (a) The transformer-based HBF architecture; and (b) Sum rate achieved by different HBF schemes versus feedback overhead.

of three transformer encoder modules, as shown in Fig. 5 (a). According to [11], analog RF beamformer and digital baseband beamformer can be optimized to approach the optimal fully-digital beamformer. Motivated by this principle, each transformer encoder in Fig. 5 (a) implements a part of the HBF optimization. More specifically, the input dimension of the first transformer encoder is $K \times 2N_uN_t$, where N_u is the number of UEs, and the output represents the fully-digital beamformer, i.e., $\mathbf{F} \in \mathcal{C}^{K\times 2N_uN_t}$; the input dimension of the second transformer encoder is the permutation of $\mathbf{F}$, i.e., $N_u \times 2KN_t$, and the output is the phase of the analog RF beamformer, i.e., $\mathbf{P} = \text{vec}(\text{angle}(\mathbf{F}_{\text{RF}})) \in \mathcal{C}^{N_u\times N_t}$; the third transformer encoder represents the digital baseband beamformer, which takes $\mathbf{F}$ as input and produces $\mathbf{F}_{\text{BB}} \in \mathcal{C}^{K\times 2N_uN_u}$ as output, respectively. By introducing the structural prior information of traditional optimization methods, combined with the self-attention's feature extraction ability, we can achieve better performance than traditional as well as existing DL-based methods in the literature. As shown in Fig. 5 (a), we consider two working modes: 1) the first mode requires an estimated CSI matrix as input, which is achieved by the adopted CSI feedback scheme; 2) the second mode relies on implicit CSI as input, which is conveyed by the feedback bits transmitted from the UEs, and in this case, the CSI feedback network is jointly trained with the proposed HBF network. Note that the case in which the proposed HBF network is trained with the perfect CSI matrix as input (working mode 1), can be regarded as an upper bound for the case trained with quantized CSI feedback bits (working mode 2).

To illustrate the superior performance of our transformer-based HBF, we use the channel parameters similar to those in Subsection III-A. We set the number of UEs to $N_u = 2$. We choose three benchmarks for comparison, namely SS-HP from [11], CNN-based HBF of [12], and MLP-based HBF from [13]. The sum rate comparison of different schemes is depicted in Fig. 5 (b). It can be seen that the transformer-based HBF scheme significantly outperforms SS-HP and other DL-based HBF schemes with both complete and limited CSI feedback. The performance gains over the benchmarks are particularly considerable at low feedback overhead of 3 to 24 bits. Moreover, the proposed scheme with limited feedback bits even outperforms SS-HP with perfect CSI, when the feedback overhead is greater than 24 bits. This demonstrates the effectiveness of the proposed transformer-based HBF architecture, particularly under the practical limited feedback scenario. However, both 'Transformer' and 'Transformer-S' have higher FLOPs and runtime than other DL-based schemes. Therefore, it is of interest to develop a more efficient transformer-based HBF architecture with guaranteed performance.

D. Semantic Communication

Our communication networks have been traditionally conceived and designed as bit pipes; that is, the goal has been to deliver as many bits as possible with the highest reliability. Current communication networks do not take into account the meaning or the purpose of the delivered bits, whose interpretation and processing have been left to higher layers. To meet the requirements of 6G wireless networks, however, it is important to propose more efficient information acquisition and delivery methods. The recently growing trend of semantic communication aims at accurately recovering the statistical structure of the underlying information of the source signal and designing the communication system in an end-to-end fashion, similarly to joint source and channel (JSC) coding by taking the source semantics into account [3], [5], [14]. Fig. 6 (a) shows the general framework of a semantic communication model, where the transmitter includes a semantic encoder and a semantic-aware JSC encoder, and the receiver includes a semantic-aware JSC decoder and a semantic decoder. In general, the transmitter can perform semantic encoding on the source according to the knowledge library for obtaining highly compressed abstract semantics, followed by JSC encoder and subsequent baseband signal processing. The receiver follows the reverse steps of the transmitter, where a JSC decoder is followed by a semantic decoder based on some knowledge library. Alternatively, the semantic and JSC encoder/decoder operations can be considered into single module as in [3].

Semantic communication is particularly effective for complex information sources, such as text, speech, image, or video, where the reconstruction quality depends on the source semantics, and is often difficult to measure through traditional measures of bit error rate or mean square error. In [14], the authors proposed a LSTM-based model to extract the semantic information of sentences through JSC coding for text transmission. However, due to the lack of a separate semantic coding module, JSC coding can only implicitly utilize the

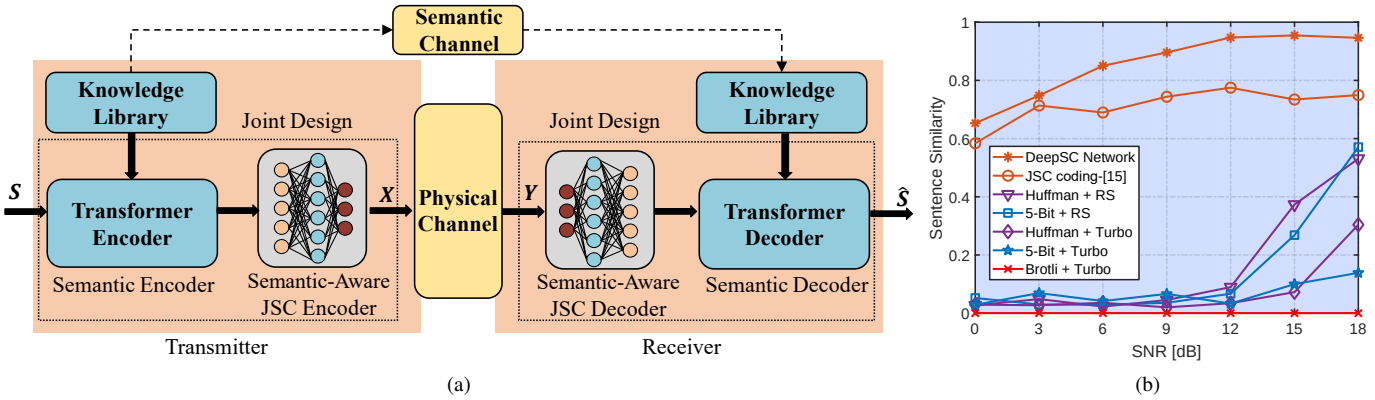


Fig. 6. (a) The transformer-based semantic communication architecture proposed in [5]; and (b) Sentence similarity of various schemes versus SNR for the same total number of transmitted symbols over the Rayleigh fading channel quoted from [5].

semantic information, which has difficulty to represent specific semantics. Instead, the transformer can extract correlations between different words to form highly abstract semantics. Inspired by this benefit, a DL-enabled semantic communication (DeepSC) scheme was proposed in [5], where a separate semantic coding network is utilized to better extract accurate semantic information. As shown in Fig. 6 (a), a transformer encoder is utilized as the semantic encoder and a MLP is used as the JSC encoder. The Rayleigh fading channel is interpreted as an untrainable layer in the model. Correspondingly, the receiver consists of a MLP-based JSC decoder followed by a transformer decoder for text reconstruction. The whole network is trained in an end-to-end fashion to simultaneously minimize the sentence similarity and maximize the mutual information. Fig. 6 (b) compares the performance of the DeepSC network [5] in transmitting text over a Rayleigh fading channel with the following benchmarks: Huffman code followed by Reed-Solomon (RS) coding and 64-quadrature amplitude modulation (QAM), fixed-length code (5-bit) followed by RS coding and 64-QAM, Huffman code followed by a Turbo coding and 64-QAM, 5-bit code followed by a Turbo coding and 128-QAM, Brotli code followed by a Turbo coding and 8-QAM, and the JSC coding approach of [14]. The simulation results demonstrate that thanks to the powerful transformer architecture, the sentence similarity performance of DeepSC [5] far outperforms all traditional approaches based on separate compression followed by channel coding, as well as the JSC coding approach [14]. Hence, we foresee that semantic-aided communication is an important challenge, where the transformer architecture is likely to have an impact on future communication systems by more effectively learning and adapting to the statistics of complex signals, such as text, image, or video.

While we have considered a simple single-input single-output channel in the example in Fig. 6, 6G communication networks will need to combine semantic communication with massive MIMO and other core communication tools and techniques, as illustrated in Fig. 2 (b). This will require jointly optimizing these modules in an end-to-end fashion. One of the challenges facing semantic communications is to achieve the potential gains from the joint processing of source, channel coding and other components while retaining the low-complexity and modular network architecture.

IV. Challenges and Open Issues

We hope that the above examples have convinced the readers of significant potentials of the transformer architecture for future 6G intelligent networks. In addition to these examples, we expect that the transformers will find applications in waveform design, channel modeling and generating, signal detection, as well as more advanced sensing techniques exploiting other complex information sources such as LiDAR or cameras. We would like to highlight that the transformer architecture was invented only in 2017. Although it has received significant attention in the last years thanks to its superior performance, the research on transformer-based communication system design is still in its infancy, and many key issues are still open. In this section, we discuss several potential directions for future study.

Network Efficiency: An important limitation of the transformer architecture is its high computation and memory complexity, mainly due to the self-attention module. This results in a significant increase in the training time and energy, and even considerable inference complexity, particularly for long sequences. This can prevent its implementation on resource-limited devices such as mobile phones. Recently, various model variants have been proposed to improve the computational and memory efficiency, such as sparse attention, linearized attention, low-rank self-attention, etc. [15]. However, existing papers focus mainly on CV and NLP applications, and there is no verification of whether the direct migration of these low-complexity solutions to wireless communications offers the same advantages. Hence, to successfully apply transformers in 6G networks, a promising research direction is to tame their complexity and memory requirements by developing highly effective and efficient transformer architectures for wireless applications.

Network Generalization: In high-speed time-varying scenarios, such as high-speed railway and low orbit satellites, the model mismatch problem is difficult to deal with due to the large target dynamic range. Moreover, since the transformer makes few assumptions on the structural bias of the input data, the network cannot perform real-time parameter retraining to address these model mismatches. One potential solution is

to utilize very large-scale data for pre-training, so that the transformer-based model can learn the knowledge covering a wide range of communication scenarios. Then the transformer-based network can be fine tuned online based on the small-scale data obtained in real time to meet the actual communication needs. Another idea can be to introduce structural biases or regularization based on prior information about the communication channel or information sources to accelerate the fine-tuning process.

Combination with Model-Driven DL: Model-driven DL methods introduce learnable parameters while retaining the model assumptions and the iterative process of model-driven methods. This, in general, results in smaller sample training and faster convergence. However, the performance can deteriorate severely when the underlying model is inaccurate. Inspired by transformer's feature extraction capability from the underlying statistics, combining transformer with model-driven DL approaches can compensate for the errors caused by inaccurate modeling. Therefore, an advanced data-model dual-driven DL architecture compatible with the advantages of model-driven optimization and the transformer architecture should be investigated.

Efficient Information Injection: In NLP, the text is divided into words, and a word embedding is used to feed each word to the transformer network. Similarly, in CV, each image is divided into patches, and the sequences of linear embedding of these patches are fed as input to a transformer. Similar techniques can be used for the semantic communication of text and image sources; however, for the physical layer design, the input is mainly based on CSI, which commonly has four dimensions: time-space-frequency-user. In this article, the inputs of channel estimation and CSI feedback take self-attention on the frequency domain of CSI, while in the hybrid beamforming, the frequency and user domains are used jointly. Therefore, how to efficiently feed the underlying input, which can include the source signal, CSI tensor, location, traffic and environment information, input the transformer architecture is one of the topics to be investigated for wireless applications.

V. Conclusions

In this article, we have presented the transformer architecture and provided examples to highlight its potential benefits in addressing various challenges for 6G intelligent networks. We have considered the applications of the transformer from massive MIMO processing to semantic communication, and provided concrete examples to show its competitive performance compared to the other classical as well as recently proposed DL-based models, hence demonstrating its great potential in designing the AI-native future communication systems. Potential research directions have also been identified to channel the efforts of the research community to the transformer-based 6G intelligent network paradigm.

References

[1] D. Gündüz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. R. Murthy, and M. van der Schaar, "Machine learning in the air," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2184-2199, Oct. 2019.

[2] Z. Qin, H. Ye, G. Y. Li, and B. F. Juang, "Deep learning in physical layer communications," *IEEE Wirel. Commun.*, vol. 26, no. 2, pp. 93-99, Apr. 2019.

[3] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 567-579, Sep. 2019.

[4] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, (Long Beach, CA, USA), Dec. 4-9, 2017, pp. 5998–6008.

[5] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663-2675, 2021.

[6] X. Ma, Z. Gao, F. Gao, and M. Di Renzo, "Model-driven deep learning based channel estimation and feedback for millimeter-wave massive hybrid MIMO systems," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2388-2406, Aug. 2021.

[7] X. Ma and Z. Gao, "Data-driven deep learning to design pilot and channel estimator for massive MIMO," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5677-5682, May 2020.

[8] M. B. Mashhadi and D. Gündüz, "Pruning the pilots: Deep learning-based pilot design and channel estimation for MIMO-OFDM systems," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 10, pp. 6315-6328, Oct. 2021.

[9] C. Lu, W. Xu, S. Jin, and K. Wang, "Bit-level optimized neural network for multi-antenna channel quantization," *IEEE Wirel. Commun. Lett.*, vol. 9, no. 1, pp. 87-90, Jan. 2020.

[10] C. Lu, W. Xu, H. Shen, J. Zhu, and K. Wang, "MIMO channel information feedback using deep recurrent network," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 188-191, Jan. 2019.

[11] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. W. Heath, "Spatially sparse precoding in millimeter wave MIMO systems," *IEEE Trans. Wirel. Commun.*, vol. 13, no. 3, pp. 1499-1513, Mar. 2014.

[12] A. M. Elbir and K. V. Mishra, "Low-complexity limited-feedback deep hybrid beamforming for broadband massive MIMO," in *Proc. IEEE 21th Int. Workshop Signal Process. Adv. Wirel. Commun.* (Atlanta, GA, USA), May 26-29, 2020, pp. 1-5.

[13] G. Zhen, M. Wu, C. Hu, F. Gao, G. Wen, D. Zheng, and J. Zhang, "Data-driven deep learning based hybrid beamforming for aerial massive MIMO-OFDM systems with implicit CSI" arXiv preprint arXiv:2201.06778, 2022.

[14] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* (Calgary, AB, Canada), Apr. 15-20, 2018, pp. 2326-2330.

[15] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient Transformers: A survey" arXiv preprint arXiv:2009.06732, 2022.