Edge-Fog Computing-Enabled EEG Data Compression via Asymmetrical Variational Discrete Cosine Transform Network

Xin Zhu, *Student Member, IEEE*, Hongyi Pan, *Member, IEEE*, and Ahmet Enis Cetin, *Fellow, IEEE*

Abstract—The large volume of electroencephalograph (EEG) data produced by brain-computer interface (BCI) systems presents challenges for rapid transmission over bandwidth-limited channels in Internet of Things (IoT) networks. To address the issue, we propose a novel multi-channel asymmetrical variational discrete cosine transform (DCT) network for EEG data compression within an edge-fog computing framework. At the edge level, low-complexity DCT compression units are designed using parallel trainable hard-thresholding and scaling operators to remove redundant data and extract the effective latent space representation. At the fog level, an adaptive filter bank is applied to merge important features from adjacent channels into each individual channel by leveraging inter-channel correlations. Then, the inverse DCT reconstructed multi-head attention is developed to capture both local and global dependencies and reconstruct the original signals. Furthermore, by applying the principles of variational inference, a new evidence lower bound is formulated as the loss function, driving the model to balance compression efficiency and reconstruction accuracy. Experimental results on two public datasets demonstrate that the proposed method achieves superior compression performance without sacrificing any useful information for BCI detection compared with state-of-the-art techniques, indicating a feasible solution for EEG data compression.

Index Terms—Brain-computer interface (BCI), edge-fog computing, EEG data compression, discrete cosine transform, multi-channel asymmetrical network, evidence lower bound.

I. INTRODUCTION

THe brain-computer interface (BCI) is emerging as a pivotal technology within the framework of Industry 4.0 [1]. BCI facilitates direct communication between the human brain and machines, bypassing traditional physical interfaces [2]. This capability holds transformative potential for enhancing automation and operational efficiency in industrial environments. For instance, BCI allows operators to control robotic arms or machinery with greater precision and less physical effort [3]. Furthermore, BCI can enhance worker safety by monitoring cognitive load and stress levels [4].
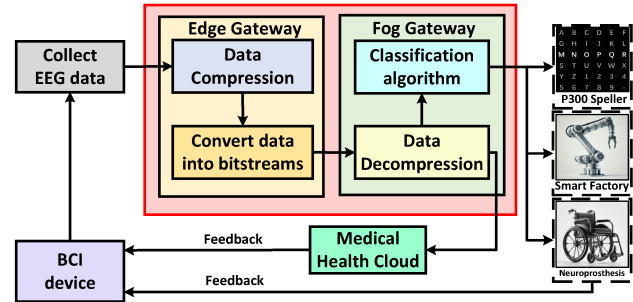
Fig. 1: Edge-fog computing framework.

The BCI system captures brain signals by measuring metabolic activity through functional near-infrared spectroscopy (fNIRS), magnetic activity using magnetoencephalography (MEG), or electrical activity via electroencephalography (EEG) [5]. EEG is recorded via electrodes placed on the scalp. It is widely utilized and well-suited for industrial applications due to its non-invasive nature, cost-effectiveness, high temporal resolution, and portability [6].

The integration of BCI into the industrial field is intricately linked with the Internet of Things (IoT) [7], [8]. In smart factories, IoT enables various devices and systems to communicate and operate cohesively. BCI-equipped operators can manage multiple IoT-connected devices simultaneously to optimize workflows. However, the vast volumes of EEG data generated by the BCI system pose significant challenges to rapid transmission and real-time analysis in bandwidth-limited channels [9]. Edge and fog computing frameworks [10] address these challenges by deploying the edge gateway close to the data generators, such as biosensors, to perform real-time data compression, as illustrated in Fig. 1. The compressed data is then transmitted to the fog gateway for decompression and feature extraction before being forwarded to BCI applications. Additionally, the reconstructed data is uploaded to the medical health cloud for storage and further analysis. Finally, both the applications and the medical cloud provide feedback to the users. The fog gateway, positioned between the cloud and the sensors, provides an intermediary layer that enhances processing capabilities and reduces reliance on cloud resources for comprehensive data analysis. In contrast, while the edge gateway is situated near sensors, it is limited in its computational capabilities, rendering it insufficient for complex tasks such as training neural network models and performing extensive analyses on large-scale EEG datasets. Therefore,

an efficient low-complexity EEG compression algorithm is required at the edge gateway to eliminate redundant data, reducing the latency, data transfer costs and bandwidth.

EEG signal compression techniques are generally divided into three main categories: traditional signal processing methods, neural network-based methods, and transform-based learning techniques [11].

Traditional signal processing methods, such as discrete cosine transform (DCT) and discrete wavelet transform (DWT), have been extensively employed in EEG data compression. These compression techniques can be categorized into single-channel and multi-channel schemes. In the single-channel methods, each EEG channel was processed independently using compression techniques, e.g., the fast DCT [12], DWT [13]. In contrast, multi-channel compression algorithms perform simultaneous compression of EEG signals across all channels by exploiting inter-channel correlations. For instance, Hejrati *et al.* utilized the K-means algorithm to cluster EEG channels [14]. Similar channels were then grouped to eliminate inter-channel dependencies, thereby reducing the data size. However, the maximum compression ratio achieved in [14] was smaller than 3, indicating a low compression efficiency.

Recently, neural networks have been increasingly utilized for data compression due to their capacity to learn complex data representations, such as the underlying patterns in EEG signals. Among these methods, autoencoders are the most frequently employed for compression tasks. For instance, the convolutional autoencoder (CAE) [15] and variational autoencoder [16] have demonstrated their efficacy in EEG data compression by applying convolutional layers, max-pooling layers, and evidence lower bound (ELBO) to optimize their models. Furthermore, Lerogeron *et. al* proposed a neural network-based approximation of dynamic time warping (DTW) as a reconstruction loss function, enhancing the ability of convolutional autoencoders to compress EEG signals while preserving essential information [17]. Nevertheless, these autoencoder-based models capture effective latent representations of EEG signals by incorporating numerous convolutional layers or linear layers, leading to increased computational complexity and higher hardware costs.

Over the past few decades, transform-based learning methods have gained considerable attention in EEG data compression for their ability to reduce data size while maintaining critical signal information. The authors in [18] proposed a learning-based adaptive transform method, which combines the DCT with an artificial neural network (ANN) to achieve near-lossless compression. The DCT is utilized to compress the original data, and then the MLP further compresses the primary DCT coefficients. Additionally, an asymmetrical sparse autoencoder [11] was designed for compressing EEG sensor signals. This model integrates a low-complexity DCT layer within the encoder module to eliminate redundant coefficients and generate the latent representation. In contrast, the decoder module is equipped with additional linear layers to improve the quality of signal reconstruction. Nonetheless, These models function as single-channel approaches, where EEG signals are compressed and reconstructed on a per-channel basis, neglecting the correlations that exist among multi-channel EEG

signals. As a result, the reconstruction accuracy is limited.

To address the aforementioned issues, this paper proposes a novel multi-channel asymmetrical variational discrete cosine transform network (AVDCT-Net) to compress EEG signals across all channels simultaneously within the edge-fog computing framework. At the edge gateway, each channel is processed through a linear layer, followed by a low-complexity DCT compression unit (DCU) in the encoder module. In this unit, parallel hard-thresholding nonlinearities eliminate redundant data, while multiple scaling operators serve as an approximate filter bank within the DCT domain. The entire compression process is guided by an ELBO-based cost function, enforcing a regularization constraint on the latent space coefficients to ensure a more structured and efficient representation. At the fog gateway, the decoder module reconstructs the EEG signal from compressed data utilizing an adaptive filter bank, inverse DCT (IDCT) reconstructed multi-head attention (IRMHA), and linear layers. The key contributions of this paper are summarized as follows:

1) We propose a novel edge-fog computing-enabled multi-channel AVDCT-Net model to perform EEG data compression. By introducing the DCU to the encoder and employing the newly derived ELBO as an objective loss function, the AVDCT-Net achieves the trade-off between compression efficiency and reconstruction fidelity. Additionally, the encoder module exhibits low complexity, making it well-suited for deployment on edge gateways with limited computational resources.

2) An adaptive filter bank is implemented to promote the incorporation of important features from neighboring channels into each individual channel by leveraging the inter-channel correlations. This adaptive integration enhances reconstruction accuracy by compensating for any potential information loss that may occur during compression.

3) We design a new IRMHA based on the multi-head attention (MHA), hard-thresholding operator, and IDCT. The MHA mechanism strengthens the decoder's feature extraction by capturing both local and global dependencies, while the IDCT aids in accurately reconstructing the original data.

4) Evaluation results demonstrate that the proposed method surpasses other state-of-art compression models in terms of compression efficiency and reconstruction accuracy on two EEG datasets: the BCI2 dataset [19] and the BCI3 dataset [20]. Moreover, it is experimentally shown that the proposed method achieves effective EEG signal compression without compromising any important information for the BCI detection system.

II. PRELIMINARIES

A. The Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) [21] is extensively employed in data compression, particularly within established standards such as JPEG [22] for image compression and MPEG [23] for video compression. Similar to the discrete Fourier transform, the DCT transforms a signal from the time domain to the frequency domain, allowing the data to be represented as a sum of cosine functions oscillating at various

frequencies. Given a sequence $\mathbf{x}$ of length L, the orthogonal type DCT-III is computed as follows:

$$X_i = \sqrt{\frac{1}{L}} x_0 + \sqrt{\frac{2}{L}} \sum_{l=1}^{L-1} x_l \cos\left[\frac{\pi}{L}\left(i + \frac{1}{2}\right) l\right], \quad (1)$$

where $0 \le i \le L-1$. The inverse DCT (IDCT) is defined as:

$$x_n = \alpha_n \sum_{i=0}^{L-1} X_i \cos\left[\frac{\pi}{L}\left(i + \frac{1}{2}\right) n\right], n = 0, \ldots, L-1, \quad (2)$$

where $\alpha_n = \sqrt{\frac{1}{L}}$ if $n = 0$, otherwise $\alpha_n = \sqrt{\frac{2}{L}}$.

The DCT is effective at concentrating the majority of signal energy into a limited number of low-frequency components. This capability allows for the preservation of critical information while enabling the elimination of less significant high-frequency data.

B. The Variational Autoencoder (VAE)

A variational autoencoder (VAE) [24] is a generative model that integrates principles from deep learning and Bayesian inference. Its application spans a wide range of tasks, including image synthesis, noise reduction and feature classification. In the VAE formulation, the observed data $\mathbf{x}$ are presumed to be generated by an underlying stochastic process, which involves an unobserved latent variable $\mathbf{z}$ and is parameterized by ϑ. Its marginal log-likelihood can be expressed as:

$$\log p_\vartheta(\mathbf{x}) = \log \int p_\vartheta(\mathbf{x}, \mathbf{z}) \, d\mathbf{z}. \quad (3)$$

Direct computation of this integral is often intractable due to the complexity of the joint distribution $p_\vartheta(\mathbf{x}, \mathbf{z})$. To solve this problem, an approximate posterior distribution $q_\varphi(\mathbf{z}|\mathbf{x})$ is introduced to serve as the **encoder**, where φ represents the weights. After that, Eq. (3) is rewritten as:

$$\log p_\vartheta(\mathbf{x}) = \log \int q_\varphi(\mathbf{z}|\mathbf{x}) \frac{p_\vartheta(\mathbf{x}, \mathbf{z})}{q_\varphi(\mathbf{z}|\mathbf{x})} \, d\mathbf{z}. \quad (4)$$

Next, applying Jensen inequality to Eq. (4) yields the Evidence Lower Bound (ELBO) $\mathcal{L}(\mathbf{x})$:

$$\log p_\vartheta(\mathbf{x}) \ge \mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\vartheta(\mathbf{x}, \mathbf{z})}{q_\varphi(\mathbf{z}|\mathbf{x})}\right] = \mathcal{L}(\mathbf{x}). \quad (5)$$

Given that $p_\vartheta(\mathbf{x}, \mathbf{z}) = p_\vartheta(\mathbf{x}|\mathbf{z}) p_\vartheta(\mathbf{z})$, the ELBO can be further decomposed as:

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}\left[\log p_\vartheta(\mathbf{x}|\mathbf{z})\right] + \mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\vartheta(\mathbf{z})}{q_\varphi(\mathbf{z}|\mathbf{x})}\right], \quad (6)$$

where $p_\vartheta(\mathbf{x}|\mathbf{z})$ models the reconstruction **decoder** and its expected log-likelihood quantifies the reconstruction loss. Additionally, the second right-hand side term corresponds to the negative Kullback-Leibler divergence $-D_{KL}(q_\varphi(\mathbf{z}|\mathbf{x})\|p_\vartheta(\mathbf{z}))$. It promotes a well-regularized latent space by penalizing deviations of the approximate posterior $q_\varphi(\mathbf{z}|\mathbf{x})$ from the prior distribution $p_\vartheta(\mathbf{z})$. Therefore, Eq. (6) can rewritten as:

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}\left[\log p_\vartheta(\mathbf{x}|\mathbf{z})\right] - D_{KL}(q_\varphi(\mathbf{z}|\mathbf{x})\|p_\vartheta(\mathbf{z})). \quad (7)$$

It is noticed that Eq. (7) is tractable and can be optimized efficiently. Hence, ELBO is a surrogate objective function when

directly maximizing $\log p_\vartheta(\mathbf{x})$ is computationally prohibitive. Moreover, maximizing the ELBO involves two key objectives: improving data reconstruction accuracy via the expected log-likelihood term and regularizing the latent space through the KL divergence term.

III. METHODOLOGY

This section presents the proposed multi-channel asymmetrical variational discrete cosine transform network (AVDCT-Net) for EEG data compression. The network employs an encoder with low computational complexity, making it suitable for deployment on a resource-constrained edge gateway. In contrast, the decoder is designed with a more intricate structure to improve the accuracy of data reconstruction. This asymmetry is feasible because the decoder runs on a more capable fog gateway, allowing for more intensive computational processes than the edge gateway [10].

A. Asymmetrical Variational Discrete Cosine Transform Network (AVDCT-Net)

In EEG systems, signals from all channels are captured simultaneously via scalp electrodes within the EEG headset. To maintain synchronization and minimize delays or data backlogs, AVDCT-Net leverages a multi-channel framework as shown in Fig. 2, enabling the concurrent compression of signals across all channels. In this study, we employ 64-channel EEG signals due to their widespread use in BCI experiments for classification and detection tasks [19], [20], [25]. Additionally, the collected signal from each channel is segmented into short-time blocks of length L.

1) Encoder module of the AVDCT-Net at the edge gateway: As shown in Fig. 2, the input block $\mathbf{x}_c \in \mathbb{R}^L$ is initially processed by the c-th linear layer to extract the features, producing the filtered output $\bar{\mathbf{x}}_c \in \mathbb{R}^L$, where $0 \le c \le C-1$. Here, $C = 64$ denotes the total number of channels. Next, a multi-channel DCT compression block is designed to perform EEG data compression, as shown in Fig. 3. For each unit in the block, the DCT is employed to transform EEG signals from the time domain to the frequency domain using Eq. (1). The resulting transformed signal, $\hat{\mathbf{x}}_c \in \mathbb{R}^L$, is then passed through N subbands. Each subband is equipped with a trainable hard-thresholding operator and a scaling operator. The hard-thresholding operator is applied to introduce nonlinearity between the DCT and scaling operator. Additionally, it can also eliminate small entries, which are typically redundant information and noise in the DCT domain. It is defined as:

$$\tilde{\mathbf{x}}_{c,n} = \mathcal{R}_{c,n}(\hat{\mathbf{x}}_c) + \mathbf{t}_{c,n} \cdot \text{sign}(\mathcal{R}_{c,n}(\hat{\mathbf{x}}_c)), \quad (8)$$

where

$$\mathcal{R}_{c,n}(\hat{\mathbf{x}}_c) = \text{sign}(\hat{\mathbf{x}}_c) \cdot (|\hat{\mathbf{x}}_c| - \mathbf{t}_{c,n})_+ \quad (9)$$

is the soft-thresholding operator [26]. $\mathbf{t}_{c,n}$ represents the threshold for $0 \le n \le N-1$, which is trained using the back-propagation algorithm. $(\cdot)_+$ stands for the rectified linear unit (ReLU) function. While the soft-thresholding operator can be employed to denoise data, it may diminish the energy of
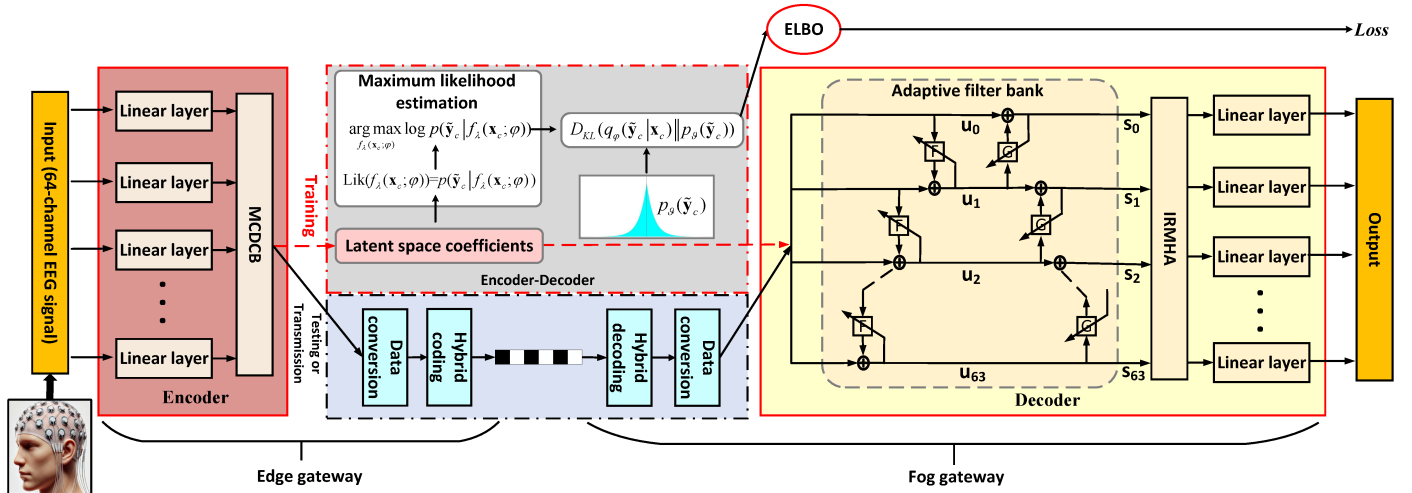
Fig. 2: Block diagram of the asymmetrical variational discrete cosine transform network.
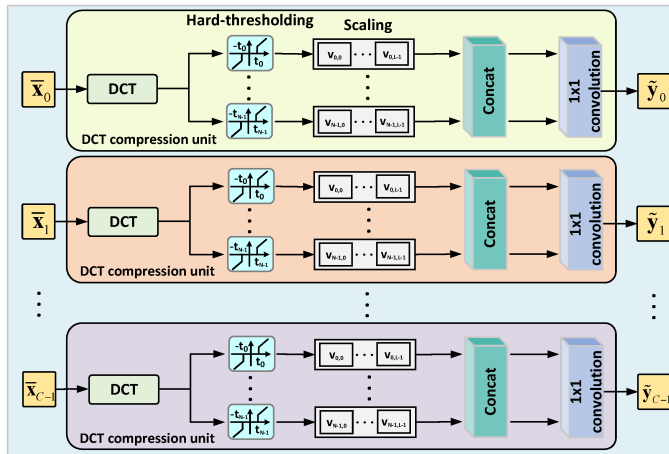

Fig. 3: Multi-channel DCT compression block (MCDCB).

Algorithm 1: DCT compression unit

Input: $\mathbf{x} \in \mathbb{R}^L$

Output: $\mathbf{z} \in \mathbb{R}^L$

Define: $\mathbf{t}_n \in \mathbb{R}^L, \mathbf{V}_n \in \mathbb{R}^L, \mathbf{Conv} = \text{Conv1D}(\text{in} = N, \text{out} = 1, \text{ kernel size} = 1)$;

$\mathbf{y} = \mathbf{DCT}(\mathbf{x}) \in \mathbb{R}^L$;

for $n = 0; n \leq N - 1; n = n + 1$ **do**

$\quad \mathbf{e}_n = \text{sign}(\mathbf{y}_n)(|\mathbf{y}_n| - \mathbf{t}_n)_+ \in \mathbb{R}^L$;

$\quad \tilde{\mathbf{e}}_n = \mathbf{e}_n + \mathbf{t}_n \circ \text{sign}(\mathbf{e}_n) \in \mathbb{R}^L$;

$\quad \mathbf{h}_n = \tilde{\mathbf{e}}_n \circ \mathbf{V}_n \in \mathbb{R}^L$;

end

$\mathbf{h} = \text{Concat}(\mathbf{h}_n) \in \mathbb{R}^{N \times L}$;

$\mathbf{z} = \mathbf{Conv}(\mathbf{h}) \in \mathbb{R}^L$;

return $\mathbf{z} \in \mathbb{R}^L$;

---

prominent DCT coefficients, resulting in errors during data reconstruction.

In contrast to the manually designed quantization matrix employed in the JPEG standards, our model incorporates the trainable scaling vector that automatically optimizes the weighting of each DCT coefficient during the training process. The computation is defined as:

$$\mathbf{y}_{c,n} = \tilde{\mathbf{x}}_{c,n} \circ \mathbf{v}_{c,n}, \tag{10}$$

where $\circ$ stands for element-wise product. The scaling vector $\mathbf{v}_{c,n} \in \mathbb{R}^L$ is learned via the back-propagation algorithm.

Another perspective on the scaling layer pertains to the DCT convolution theorem i.e., symmetric convolution in the time domain can be achieved through element-wise multiplication in the DCT domain [27]. It is expressed as:

$$\mathbf{x} \otimes \mathbf{w} = \mathcal{D}^{-1}(\mathcal{D}(\mathbf{x} \circ \mathbf{a}) \circ \mathcal{D}(\mathbf{w} \circ \mathbf{a})) \circ \mathbf{k}, \tag{11}$$

where $\mathbf{x} \in \mathbb{R}^L$ and $\mathbf{w} \in \mathbb{R}^L$ are two vectors. $\mathcal{D}(\cdot)$ and $\mathcal{D}^{-1}(\cdot)$ denote the orthogonal type-III DCT and IDCT, respectively.

$\otimes$ stands for the symmetric convolution [28]. $\mathbf{a}$ represents a constant vector:

$$\mathbf{a}[i] = \begin{cases} 1/(2\sqrt{L}), & i = 0, \\ \sqrt{1/(2L)}, & i > 0, \end{cases} \tag{12}$$

where $0 \leq i \leq L - 1$. $\mathbf{k}[i] = 1/\mathbf{a}[i]$. Hence, each scaling layer in the DCT domain is akin to a trainable filter, as it can be translated into symmetric convolution within the time domain. Furthermore, the combination of N scaling subbands constitutes a filter bank. It is experimentally shown in section IV-G that a combination filter bank leads to the improvement of feature extraction capability and compression performance. In addition, the proposed model achieves optimal compression results when $N = 3$.

After the scaling layer, the coefficients across all subbands are concatenated to form $\widehat{\mathbf{Y}}_c \in \mathbb{R}^{N \times L}$. Then, the one-by-one convolutional layer is introduced to reduce the dimensionality of $\widehat{\mathbf{Y}}_c$ from $\mathbb{R}^{N \times L}$ to $\mathbb{R}^L$, which increases the data compression efficiency in the latent space. Assume the convolutional layer output is $\tilde{\mathbf{y}}_c \in \mathbb{R}^L$. The overall operation in each DCT compression unit is outlined in the Algorithm 1.
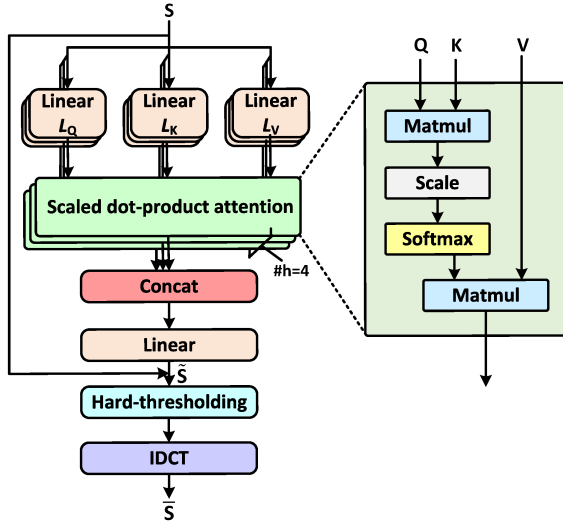
Fig. 4: Inverse DCT reconstructed multi-head attention.

2) Decoder part of the AVDCT-Net at the fog gateway:
After receiving the data from the encoder, the decoding process begins with the application of the adaptive filter bank. In multi-channel EEG systems, the signals recorded across different channels are often correlated due to the distributed nature of brain activity [29]. However, previous neural network-based compression models almost ignored such inter-channel correlations, which may provide important information for reconstruction. Therefore, the adaptive filter bank is employed to facilitate the integration of salient features from adjacent channels into each EEG channel. In this framework, the trainable scaling vectors $\mathbf{g}_c$ and $\mathbf{f}_c$ are employed as adaptive filters, based on the principle that scaling in the DCT domain can be equivalently transformed into convolution in the time domain. The computation is expressed as:

$$\mathbf{u}_c = \begin{cases} \tilde{\mathbf{y}}_0, & c = 0, \\ \tilde{\mathbf{y}}_c + \tilde{\mathbf{y}}_{c-1} \circ \mathbf{f}_{c-1}, & 1 \leq c \leq C-1, \end{cases} \quad (13)$$

$$\mathbf{s}_c = \begin{cases} \mathbf{u}_c + \mathbf{u}_{c+1} \circ \mathbf{g}_c, & 0 \leq c \leq C-2, \\ \mathbf{u}_c, & c = C-1, \end{cases} \quad (14)$$

where $\tilde{\mathbf{y}}_c$ is the encoder output. $\mathbf{u}_c$ and $\mathbf{s}_c$ represent the outputs of the first and second filter banks, respectively, as shown in Fig 2. During the lossy compression process, some critical information may be compromised. However, by employing an adaptive filter bank, each channel can recover potentially lost valuable information from correlated channels.

The adaptive filter bank output can be represented as a feature map $\mathbf{S} \in \mathbb{R}^{L \times C}$. In multi-channel EEG signals, spatially adjacent channels show inter-channel correlations, while each channel also has intra-channel correlations after applying the adaptive filter bank. Therefore, the IDCT reconstructed multi-head attention (IRMHA) is designed to capture a comprehensive range of local and global dependencies from the feature map for original signal reconstruction. IRMHA comprises the multi-head attention (MHA), hard-thresholding operator, and IDCT. Specifically, $\mathbf{S}$ is first processed through three separate

linear transformations for each head. The transformations are described as:

$$\mathbf{Q}_i = \mathbf{S}\mathbf{W}_i^Q + \mathbf{b}_Q, \mathbf{K}_i = \mathbf{S}\mathbf{W}_i^K + \mathbf{b}_K, \mathbf{V}_i = \mathbf{S}\mathbf{W}_i^V + \mathbf{b}_V, \quad (15)$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{C \times D}$, $\mathbf{W}_i^K \in \mathbb{R}^{C \times D}$, and $\mathbf{W}_i^V \in \mathbb{R}^{C \times D}$ stand for the transformation matrices that map the input dimensions to a lower-dimensional space specific to each head. i indexes the head number. $D = \frac{C}{h}$. $h = 4$ represents the number of heads. $\mathbf{b}_Q \in \mathbb{R}^D$, $\mathbf{b}_K \in \mathbb{R}^D$, and $\mathbf{b}_V \in \mathbb{R}^D$ denote trainable bias. Then, each head applies the scaled dot-product attention mechanism independently:

$$\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{D}}\right)\mathbf{V}_i. \quad (16)$$

Next, the outputs of each head are concatenated and linearly transformed into the original input dimensionality:

$$\widetilde{\mathbf{S}} = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_\beta)\mathbf{W}^O, \quad (17)$$

where $\mathbf{W}^O \in \mathbb{R}^{C \times C}$ is the transform weight matrix. $\widetilde{\mathbf{S}} \in \mathbb{R}^{L \times C}$ is MHA output. As the neural network deepens, it becomes increasingly susceptible to the vanishing gradient problem. To mitigate this issue, a residual connection is implemented, directly linking the input to the output of the MHA as shown in Fig. 4. Subsequently, the hard-thresholding operator, as defined in Eq. (8), is used to eliminate redundant DCT coefficients and introduce nonlinearity after the linear layer. The IDCT of the hard-thresholded output is then computed using Eq. (2), producing the result $\bar{\mathbf{S}} \in \mathbb{R}^{L \times C}$. Finally, the reconstructed C-channel EEG signal, represented as $\mathbf{Z} \in \mathbb{R}^{L \times C}$, is obtained using C linear layers.

B. The Evidence Lower Bound (ELBO) for AVDCT-Net

To optimize the compression ratio, it is a common practice to impose a sparsity constraint on the latent space coefficients [11], [30]. While this approach mitigates redundancy, it also penalizes the primary coefficients, potentially leading to the loss of important information. Moreover, larger positive coefficients incur greater penalties according to the sparsity penalty definition [31]. To more effectively regularize the latent space and improve reconstruction accuracy, we derive a new ELBO for the multi-channel DCT compression block. In our model, Eq. (7) can be rewritten as follows:

$$\mathcal{L}(\mathbf{x}_c) = \mathbb{E}_{q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c)}\left[\log p_\vartheta(\mathbf{x}_c|\tilde{\mathbf{y}}_c)\right] - D_{KL}(q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c)\|p_\vartheta(\tilde{\mathbf{y}}_c)), \quad (18)$$

where $q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c)$ represents the approximate posterior distribution of the latent space DCT coefficients $\tilde{\mathbf{y}}_c$ conditioned on the observed data $\mathbf{x}_c$. $p_\vartheta(\mathbf{x}_c|\tilde{\mathbf{y}}_c)$ stands for the likelihood of reconstructing the original input $\mathbf{x}_c$ from its encoded representation $\tilde{\mathbf{y}}_c$. $p_\vartheta(\tilde{\mathbf{y}}_c)$ is the prior distribution of $\tilde{\mathbf{y}}_c$.

In our case, the actual distribution of $\tilde{\mathbf{y}}_c$ is unknown. However, the distribution of each DCT coefficient can be mathematically analyzed as a Laplacian distribution characterized by a location parameter $\mu = 0$ and a scale parameter λ [32]. Additionally, after the hard-thresholding and scaling operators, smaller, correlated variables are eliminated. The remaining larger variables represent distinct, essential signal

components, which can be treated as independent. As a result, each latent space coefficient can be assumed to be independently and identically distributed in our model. Hence, we represent $q_\varphi(\tilde{y}_{c,l}|\mathbf{x}_c)$ using a Laplacian distribution, for $0 \le l \le L-1$. Furthermore, the probability density function (PDF) of $q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c)$ can be defined as:

$$p(\tilde{\mathbf{y}}_c|f_\lambda(\mathbf{x}_c;\varphi)) = (\frac{1}{2f_\lambda(\mathbf{x}_c;\varphi)})^L \exp\left(-\frac{||\tilde{\mathbf{y}}_c||}{f_\lambda(\mathbf{x}_c;\varphi)}\right) \quad (19)$$

where the scale function $f_\lambda(\mathbf{x}_c;\varphi)$ is parametrized by encoder neural network with parameters φ. $||\cdot||$ denotes the 1-norm.

Assume $p(\tilde{y}_{c,l})$ is the PDF of $q_\varphi(\tilde{y}_{c,l}|\mathbf{x}_c)$. Fig. 5 illustrates the curves of $p(\tilde{y}_{c,l})$ corresponding to five different possible values of $f_\lambda(\mathbf{x}_c;\varphi)$. It is observed that when the value of $f_\lambda(\mathbf{x}_c;\varphi)$ decreases, the curve of $p(\tilde{y}_{c,l})$ becomes narrower and more peaked at zero, indicating a higher probability for zero. Additionally, the decay rate of the tails increases, resulting in a lower probability of extreme values. In EEG data compression, the promotion of sparsity within the latent space is pivotal for improving the compression ratio. Therefore, the value of $f_\lambda(\mathbf{x}_c;\varphi)$ is expected to be low. As mentioned the section II-B, the prior distribution $p_\vartheta(\tilde{\mathbf{y}}_c)$ acts as a regularizer by constraining $q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c)$ to be close to a predefined distribution. Thus, we choose $p_\vartheta(\tilde{\mathbf{y}}_c)$ as a Laplacian distribution with a small scale parameter λ. Its PDF is represented as:

$$p_\vartheta(\tilde{\mathbf{y}}_c) = (\frac{1}{2\lambda})^L \exp\left(-\frac{||\tilde{\mathbf{y}}_c||}{\lambda}\right). \quad (20)$$

After that, the term $D_{KL}(q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c)\|p_\vartheta(\tilde{\mathbf{y}}_c))$ in Eq. (18) can be calculated as follows:

$$\begin{aligned} &D_{KL}(q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c)\|p_\vartheta(\tilde{\mathbf{y}}_c)) \\ &= \int q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c) \log \frac{q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c)}{p_\vartheta(\tilde{\mathbf{y}}_c)} d\tilde{\mathbf{y}}_c \\ &= \frac{f_\lambda(\mathbf{x}_c;\varphi)L}{\lambda} + L \log \frac{\lambda}{f_\lambda(\mathbf{x}_c;\varphi)} - L. \end{aligned} \quad (21)$$

However, the exact value of $f_\lambda(\mathbf{x}_c;\varphi)$ is still unknown. Next, we utilize maximum likelihood estimation to approximately solve $f_\lambda(\mathbf{x}_c;\varphi)$ based on the observed latent space coefficients $\tilde{\mathbf{y}}_c = [\tilde{y}_{c,1},\ldots,\tilde{y}_{c,l},\ldots,\tilde{y}_{c,L}]$. The likelihood of $f_\lambda(\mathbf{x}_c;\varphi)$ is computed as:

$$\begin{aligned} \text{Lik}(f_\lambda(\mathbf{x}_c;\varphi)) &= p(\tilde{\mathbf{y}}_c|f_\lambda(\mathbf{x}_c;\varphi)) \\ &= \prod_{l=0}^{L-1} \frac{1}{2f_\lambda(\mathbf{x}_c;\varphi)} \exp\left(-\frac{|\tilde{y}_{c,l}|}{f_\lambda(\mathbf{x}_c;\varphi)}\right). \end{aligned} \quad (22)$$

To find the maximum likelihood estimate, we solve the following optimization problem:

$$\underset{f_\lambda(\mathbf{x}_c;\varphi)}{\arg\max} \log p(\tilde{\mathbf{y}}_c|f_\lambda(\mathbf{x}_c;\varphi)) = \frac{1}{L}\sum_{l=0}^{L-1}|\tilde{y}_{c,l}|. \quad (23)$$

Then, Eq (21) is rewritten as follows:

$$D_{KL} = \frac{\sum_{l=0}^{L-1}|\tilde{y}_{c,l}|}{\lambda} + L\log\lambda - L\log\frac{1}{L}\sum_{l=0}^{L-1}|\tilde{y}_{c,l}| - L. \quad (24)$$
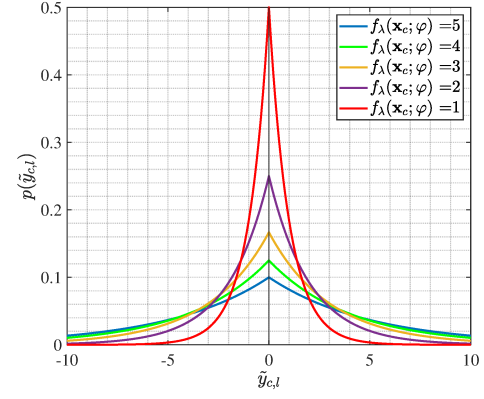

Fig. 5: $p(\tilde{y}_{c,l})$ with five different $f_\lambda(\mathbf{x}_c;\varphi)$.

It is noticed that D_{KL} achieves the minimum value when $\sum_{l=0}^{L-1}|\tilde{y}_{c,l}| = \lambda L$. Unlike the approach in [11], which penalizes each coefficient towards zero, our model constrains the sum of the latent space coefficients to a small nonzero value, λL, thereby increasing the likelihood of retaining the important DCT coefficients while eliminating the redundant ones. As λ approaches zero, D_{KL} may exhibit numerical instability. To mitigate potential overflow, $D_{KL}(q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c)\|p_\vartheta(\tilde{\mathbf{y}}_c))$ can also be replaced with $D_{KL}(p_\vartheta(\tilde{\mathbf{y}}_c)\|q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c))$. This substitution helps prevent numerical issues and promote sparsity in the learned representations.

As mentioned in section II-B, the first right-hand term in Eq. (18) represents the reconstruction loss. Here we use mean square error as reconstruction loss, which is calculated as:

$$\mathbb{E}_{q_\varphi(\tilde{\mathbf{y}}_c|\mathbf{x}_c)}\left[\log p_\vartheta(\mathbf{x}_c|\tilde{\mathbf{y}}_c)\right] = \frac{1}{L}\sum_{l=0}^{L-1}(x_{c,l}-z_{c,l})^2, \quad (25)$$

where $x_{c,l}$ and $z_{c,l}$ represent the input and output element, respectively, $0 \le c \le C-1$, $0 \le l \le L-1$.

Consequently, the new ELBO is described as:

$$\mathcal{L}_n = -\frac{1}{L}\sum_{l=0}^{L-1}(x_{c,l}-z_{c,l})^2 - \varepsilon D_{KL}, \quad (26)$$

where ε is the weight of the KL-divergence term. The training objective is to maximize the ELBO. Additionally, we utilize a threshold parameter ρ to adjust the compression ratio during the training stage. When the percentage of zeros in $\tilde{\mathbf{y}}_c$ drops below the defined threshold ρ, the training procedure is stopped. In this process, small coefficients, considered redundant, are removed by being set to zero.

C. Data Encoding and Storage

The data transmission part is comprised of data conversion and hybrid coding mechanisms, as depicted in Fig. 2. The hybrid coding strategy is constructed by integrating Run-Length Encoding (RLE) [33] with the Lempel–Ziv–Markov chain algorithm (LZMA) [34] to enhance data compression efficiency. Given that a double-precision floating-point number takes up 64 bits of memory, whereas an integer utilizes only 32 bits [35], the conversion of floating-point numbers to

integers is employed to optimize memory storage and enhance transmission rates. The conversion process is executed as follows:

$$\bar{\mathbf{y}}_c = \text{Round}(10^\tau \times \tilde{\mathbf{y}}_c/\omega), \tag{27}$$

where τ is an integer, and ω is the scaling parameter utilized to adjust the compression ratio. $\text{Round}(\cdot)$ stands for the integer rounding function. By utilizing RLE and LZMA compression techniques, the EEG signal is encoded into a bitstream, which is then prepared for wireless transmission. The entire process begins with RLE, which efficiently removes sequential zero redundancies in the latent space. Subsequently, LZMA is applied to the RLE-processed data to achieve further compression. Upon decoding, this compressed bitstream is reverted to its original integer form, after which the AVDCT-Net decoder reconstructs the signal on the fog gateway.

IV. EXPERIMENTAL RESULTS

A. Performance Metrics

In this section, to evaluate the compression performances of different data compression algorithms, we analyze metrics such as compression ratio (CR), percent root mean square difference (PRD), normalized percent root mean square difference (PRDN), and quality score (QS) [30]. They are defined as follows:

- CR is determined by dividing the original data size Q_r by the compressed data size Q_c, providing a crucial metric for evaluating the effectiveness of different algorithms in reducing data size:

$$\text{CR} = \frac{Q_r}{Q_c}. \tag{28}$$

  A higher CR demonstrates that the model is more proficient at minimizing data size.

- PRD evaluates the accuracy of data reconstruction in compression algorithms. It measures the discrepancy between the original data e_m^o and the reconstructed data e_m^r, expressed as a percentage:

$$\text{PRD} = \sqrt{\frac{\sum_{m=1}^{M}\left(e_m^o - e_m^r\right)^2}{\sum_{m=1}^{M}\left(e_m^o\right)^2}} \times 100, \tag{29}$$

  where M represents the length of the data. A lower PRD value indicates that the reconstructed data closely matches the original data, signifying higher fidelity.

- PRDN represents the normalized form of PRD. It is defined as:

$$\text{PRDN} = \sqrt{\frac{\sum_{m=1}^{M}\left(e_m^o - e_m^r\right)^2}{\sum_{m=1}^{M}\left(e_m^o - \bar{e}\right)^2}} \times 100, \tag{30}$$

  where $\bar{e}$ denotes the mean of the original signal. PRDN provides a robust measure of similarity that is less sensitive to mean shifts in the original signal than PRD. A lower PRDN value signifies superior model performance.

- QS assesses the overall performance of data compression algorithms. It integrates multiple performance aspects, such as compression ratio and reconstruction accuracy:

$$\text{QS} = \frac{\text{CR}}{\text{PRD}}. \tag{31}$$

  A higher QS reflects that the algorithm excels in effectively reducing data size while maintaining high fidelity.

B. Selected Dataset and Comparison Approaches

In this study, the performances of different compression algorithms are evaluated on the BCI2 [19] and BCI3 [20] datasets: 1) The BCI2 (Dataset-IIa) dataset provides the EEG recordings for 10 30-minute sessions from 3 subjects (A/B/C). They were collected from 64 electrodes (64-channel) placed on the scalp with a sampling frequency of 160 Hz. To evaluate the compression performances of different algorithms, we utilize these 10 sessions from subject A, labeled sequentially from "AA001" to "AA010". 2) The BCI3 (Dataset-II) dataset was recorded using Farwell and Donchin's interface. It is composed of 4 datasets: The recordings designated as "Subject_A_Train" (ATr) and "Subject_A_Test" (ATe) were acquired from subject A, whereas the recordings labeled "Subject_B_Train" (BTr) and "Subject_B_Test" (BTe) were obtained from subject B. The training and testing datasets comprise 85 and 100 64-channel EEG recordings, respectively. Each recording comprises 15 rounds of repeated stimuli. In each round, the intensification period was 100 milliseconds (ms), and the inter-stimulus interval was 75 ms. Therefore the passing time for one recording is calculated as $(100+75)\times12=2,100$ ms. With 15 repetitions, the total time amounts to 31,500 ms. The sampling frequency on the BCI3 dataset is 240 Hz. Hence, each recording contains 7,560 data points for a single channel.

To verify the efficiency of the proposed data compression algorithm, we conduct a comparison with the asymmetrical sparse autoencoder with a DCT layer (ASAEDCT) [11], which has demonstrated excellent compression performance. Besides, a state-of-the-art EEG compression method based on an Edge-Fog computing architecture is selected as a benchmark for comparison [9]. It is developed using K-means clustering and Huffman encoding (KCHE). To enhance the compression ratio, we refine KCHE by converting floating-point numbers into integers prior to transmission, as described in Eq. (27). In addition, we utilize the DCT-Perceptron [36] and VAE [37] for further comparison, given that they employ low-complexity encoders.

C. Implementation Details

The proposed model is implemented in Python on a PC equipped with an Intel Core i7-12700H CPU (4.7 GHz) and 16 GB of RAM. The AdamW optimizer [38] is applied to train the model. To achieve a tradeoff between compression ratio and reconstruction accuracy, λ and ρ are chosen as 1e-5 and 0.6, respectively. In addition, the batch size and the learning rate are set as 16 and 0.001, respectively. Moreover, to reduce the trainable parameters and computation costs, we select a small block size $L = 64$ and all channels use the same training parameters by adopting parameter sharing [39].

TABLE I: Compression experiment on the BCI2 dataset.

Sub	ASAEDCT [11]				KCHE [9]				DCT-Perceptron [36]				VAE [37]				AVDCT-Net (Proposed)			
	CR	PRD	PRDN	QS	CR	PRD	PRDN	QS	CR	PRD	PRDN	QS	CR	PRD	PRDN	QS	CR	PRD	PRDN	QS
1	5.91	16.62	16.67	0.36	3.73	16.46	16.50	0.23	7.10	16.55	16.60	0.43	7.06	23.38	23.44	0.30	7.79	15.21	15.25	0.51
2	5.83	17.17	17.21	0.34	3.63	16.05	16.08	0.23	7.25	18.10	18.14	0.40	7.15	25.60	25.66	0.28	7.74	15.08	15.11	0.51
3	5.72	16.93	16.97	0.34	3.73	17.70	17.74	0.21	6.84	18.20	18.24	0.38	6.91	25.82	25.88	0.27	7.50	16.65	16.68	0.45
4	5.48	21.09	21.10	0.26	3.58	18.72	18.72	0.19	6.84	29.69	29.70	0.23	7.02	42.60	42.62	0.16	7.22	18.60	18.61	0.39
5	6.55	20.40	20.44	0.32	3.84	19.82	19.85	0.19	7.56	22.61	22.65	0.33	7.33	31.92	31.98	0.23	8.60	18.58	18.62	0.46
6	6.90	21.27	21.34	0.32	3.99	21.54	21.60	0.19	7.77	20.25	20.32	0.38	7.47	28.52	28.61	0.26	8.98	19.06	19.12	0.47
7	5.67	21.79	21.80	0.26	3.61	19.24	19.24	0.19	6.96	30.90	30.91	0.23	7.02	43.58	43.59	0.16	7.45	19.20	19.20	0.39
8	5.71	17.43	17.46	0.33	3.65	16.71	16.74	0.22	7.01	19.77	19.81	0.35	6.99	27.46	27.51	0.25	7.54	15.33	15.36	0.49
9	5.99	19.78	19.80	0.30	3.77	19.64	19.66	0.19	7.19	24.69	24.71	0.29	7.11	34.52	34.55	0.21	7.84	17.56	17.58	0.45
10	5.73	17.74	17.78	0.32	3.69	17.77	17.81	0.21	7.05	17.68	17.72	0.40	6.98	24.44	24.49	0.29	7.58	15.42	15.46	0.49
Ave	5.95	19.02	19.06	0.32	3.72	18.37	18.39	0.21	7.16	21.84	21.88	0.34	7.10	30.78	30.83	0.24	7.82	17.07	17.10	0.46

TABLE II: Transfer learning on the BCI3 dataset.

Sub	ASAEDCT [11]				KCHE [9]				DCT-Perceptron [36]				VAE [37]				AVDCT-Net(Proposed)			
	CR	PRD	PRDN	QS	CR	PRD	PRDN	QS	CR	PRD	PRDN	QS	CR	PRD	PRDN	QS	CR	PRD	PRDN	QS
ATr	11.12	9.73	9.75	1.14	6.80	8.78	8.80	0.77	10.99	10.13	10.15	1.09	10.44	11.99	12.02	0.87	12.47	8.18	8.20	1.52
ATe	10.08	12.13	12.16	0.83	7.13	12.49	12.52	0.57	10.64	13.11	13.14	0.81	10.32	15.89	15.92	0.65	11.48	10.12	10.14	1.13
BTr	8.21	9.11	9.14	0.90	6.67	10.70	10.74	0.62	10.08	10.77	10.81	0.94	9.73	11.06	11.10	0.88	10.32	6.88	6.90	1.50
BTe	10.49	9.87	9.88	1.06	6.50	8.46	8.47	0.77	10.96	9.29	9.30	1.18	10.34	11.87	11.88	0.87	11.88	7.77	7.78	1.53
Ave	9.98	10.21	10.23	0.98	6.78	10.11	10.13	0.68	10.67	10.82	10.85	1.00	10.21	12.70	12.73	0.82	11.54	8.24	8.26	1.42

D. Data Compression Experiment

In the compression experiment on the BCI2 dataset, 70% of the data from recording "AA010" was employed for model training. The remaining 9 recordings ("AA001"–"AA009") along with 30% of recording "AA010" were utilized to test the model. In summary, the training set is composed of 1,885 samples, while the testing set comprises 25,120 samples. Besides, we choose $\tau = 2$ and $\omega = 1.2$ in Eq. (27). Table I summarizes the compression results of various models across the 10 subsets of the BCI2 dataset. In comparison to ASAEDCT, AVDCT-Net achieves a lower PRD with a higher CR, demonstrating its capacity to effectively balance compression efficiency with reconstruction accuracy. The reason is that AVDCT-Net leverages a new ELBO as the loss function to train the model. This ELBO restricts the sum of the latent space coefficients to a relatively small, nonzero value instead of penalizing each coefficient towards 0. Therefore, AVDCT-Net can preserve the primary DCT coefficients while suppressing small high-frequency components. Besides, AVDCT-Net outperforms DCT-Perceptron because it applies linear layers and muti-head attention to enhance the capability to compress and restore the original signals. Moreover, compared to KCHE, AVDCT-Net achieves a reduction in average PRD from 18.37 to 17.07 (a decrease of 7.08%) and in PRDN from 18.39 to 17.10 (a decrease of 7.01%). Furthermore, it enhances the CR from 3.72 to 7.82 (a 2.10-fold increase) and improves the QS from 0.21 to 0.46 (a 2.19-fold improvement). This is because AVDCT-Net incorporates multiple trainable hard-thresholding and scaling layers to the encoder, allowing it to eliminate redundant data more effectively than KCHE. Additionally, AVDCT-Net is superior to VAE in terms of CR and PRD. While VAE integrates Gaussian processes to enhance feature

extraction, it continues to utilize a conventional autoencoder structure. Its imperfect encoder and decoder result in higher reconstruction errors than AVDCT-Net.

Table I presents the compression results of various models initialized with a single random seed. To better assess model stability, we extend our experiment by initializing compression models with five different random seeds and computing the mean and standard deviation of CR, PRD, PRDN, and QS for each subset. The results are provided in Table I of the supplementary material. For subset 3, the CR is reported as 7.48 ± 0.04. The PRD is 16.12 ± 0.31, while the PRDN is 16.15 ± 0.31. Furthermore, the QS is measured at 0.46 ± 0.01. These results exhibit low variability, implying that AVDCT-Net has strong stability. The reason is that each DCT compression unit utilizes a multi-channel architecture to capture features across different hierarchical levels and scales, facilitating a more comprehensive understanding of input EEG signals. This design mitigates dependency on specific initialization states and enhances the generalization capability. Moreover, AVDCT-Net consistently outperforms ASAEDCT, KCHE, DCT-Perceptron, and VAE, achieving the best CR, PRD, PRDN, and QS across all 10 subsets, indicating its superior compression performance.

E. Transfer Learning Experiment

To assess the generative capabilities of various models, we perform a transfer learning experiment on the BCI3 dataset. In this experiment, the model trained on datasets obtained from Subject A is evaluated on datasets acquired from a different subject, Subject B. Specifically, 70% of the data from Subject A's recording ATr was used for model training. The remaining 30% of ATr, together with the recordings ATe, BTr, and BTe, were employed for model testing. In brief, the training set is

made up of 7,246 samples, and the testing sets comprise 3,106, 12,179, 10,352, and 12,179 samples, respectively. Additionally, we select $\tau = 4$ and $\omega = 9.3$ in Eq. (27). As tabulated in Table II, AVDCT-Net demonstrates superior performance compared to other state-of-the-art approaches across all four subsets, with average CR improvements ranging from 8.15% to 70.21%, and average PRD enhancements varying between 18.50% and 35.12%. The reason is that ASAEDCT, KCHE, DCT-Perceptron, and VAE process each EEG channel independently, neglecting the inter-channel correlations. AVDCT-Net utilizes an adaptive filter bank, allowing each EEG channel to extract and integrate relevant features from adjacent channels, thereby mitigating the potential loss of critical information inherent in the compression process. Therefore, AVDCT-Net has a more robust generalization ability than other benchmark methods.

F. Visual Assessment

To comprehensively evaluate the performance of the proposed compression algorithm, we present a visual comparison of the reconstructed EEG signals in both the time and frequency domains, as illustrated in Fig. 6. In the time domain, it is observed that the differences between reconstructed and original signals are limited to a small range, indicating that AVDCT-Net effectively preserves the temporal features of the EEG data. In the frequency domain, the Fourier transform of the signals is analyzed. Results show that the frequency components of the reconstructed signal closely match those of the original signal. It demonstrates AVDCT-Net effectively preserves the primary components in the low-frequency bands while also capturing key features in the high-frequency bands.

AVDCT-Net encoder utilizes a hard-thresholding layer to eliminate small coefficients in the DCT domain, which predominantly correspond to high-frequency components. Although this process leads to information loss in the high-frequency band, the lost information consists of noise or redundant details. Therefore, their impact on signal reconstruction accuracy is minimal, as shown in Table I and Table II. Additionally, the AVDCT-Net decoder employs an adaptive filter bank to integrate salient features from adjacent channels into each EEG channel. While this mechanism enhances feature extraction, it introduces some errors in the low-frequency band. However, as depicted in Fig. 6, these errors are limited to a small range. Consequently, their impact on reconstruction quality remains minimal. Besides, they do not compromise the performance of subsequent classification tasks as shown in Table VI.

G. Ablation Experiments

In this section, we perform ablation experiments to evaluate the impact of the primary components of the proposed method, including the scaling layer, MHA, adaptive filter bank, and ELBO. Additionally, we also investigated the effect of subband quantity on model compression performance. Moreover, CR, PRD and PRDN are three critical metrics for evaluating compression performance. Therefore, τ and ω in Eq. (27) are fine-tuned to optimize CR, PRD, and PRDN for each

TABLE III: Ablation study.

Algorithm	Metrics			
	CR	PRD	PRDN	QS
No scaling layer	7.64	17.28	17.31	0.44
No MHA	7.55	17.42	17.45	0.44
No adaptive filter bank	7.69	17.26	17.30	0.45
No ELBO	7.64	17.17	17.20	0.45
AVDCT-Net (Channel=1)	7.56	17.24	17.27	0.44
AVDCT-Net (Channel=2)	7.69	17.24	17.27	0.45
AVDCT-Net (Channel=3)	**7.82**	**17.07**	**17.10**	**0.46**
AVDCT-Net (Channel=4)	7.81	17.59	17.62	0.45

method, thereby demonstrating the effectiveness of the various modules. The results of the ablation experiments on the BCI2 dataset are summarized in Table III.

1) Ablation study on scaling layer: Removing the scaling layer from the AVDCT-Net leads to an increase in the PRD from 17.07 to 17.28, representing a 1.23% rise, and a corresponding decrease in the CR from 7.82 to 7.64, amounting to a 2.30% reduction. This is due to the fact that the scaling layer in the DCT domain can be converted into the convolutional layer in the time domain. Consequently, it strengthens the encoder's ability to extract important features, leading to improved reconstruction accuracy without compromising the compression ratio.

2) Ablation study on MHA: In the absence of the MHA module within the decoder, a noticeable degradation in performance is observed, as reflected by a decreased CR, increased PRD, and a lower QS. MHA enables the simultaneous extraction of diverse features across different dimensions of the latent space. Additionally, it captures both local and global dependencies [40], promoting better generalization and scalability in multi-channel EEG data reconstruction. Therefore, the MHA is essential in optimizing reconstruction accuracy.

3) Ablation study on adaptive filter bank: When the adaptive filter bank is not present in the AVDCT-Net, the CR is decreased from 7.82 to 7.69 (1.66%), and the PRD is increased from 17.07 to 17.26 (1.11%). This is because an adaptive filter bank allows each EEG channel to merge important features from neighbouring channels. This helps restore critical information that may be lost during the lossy compression procedure. Thus, the adaptive filter bank plays a crucial role in enhancing the accuracy of data reconstruction.

4) Ablation study on ELBO: Compared to AVDCT-Net trained with the MSE loss function, AVDCT-Net optimized using the ELBO loss function exhibits an improved compression quality score as shown in Table III. By balancing the trade-off between the reconstruction loss and the KL divergence term, ELBO inherently penalizes the model for introducing unnecessary complexity in the latent space. If certain latent variables do not contribute to improving the reconstruction accuracy, the KL divergence term drives the model to eliminate these variables by constraining their distribution to match the predefined distribution. This process removes redundant coefficients, enhancing the compression efficiency.

5) Ablation study on subband quantity: Compared to AVDCT-Net with a single subband, AVDCT-Net with three
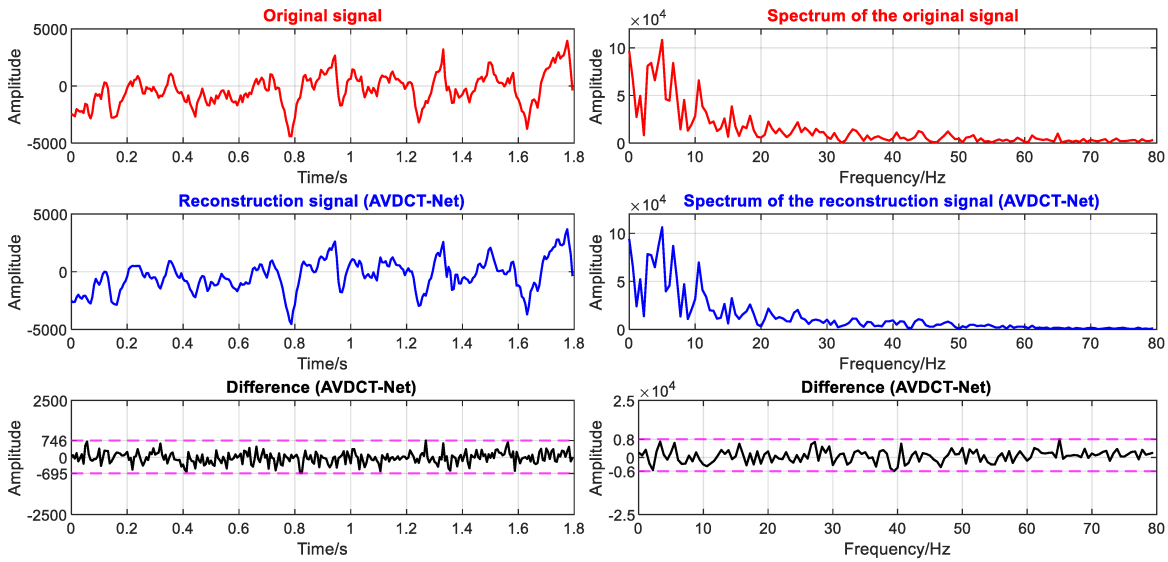
(a) Comparison between the original data and the reconstructed data on the BCI2 dataset.
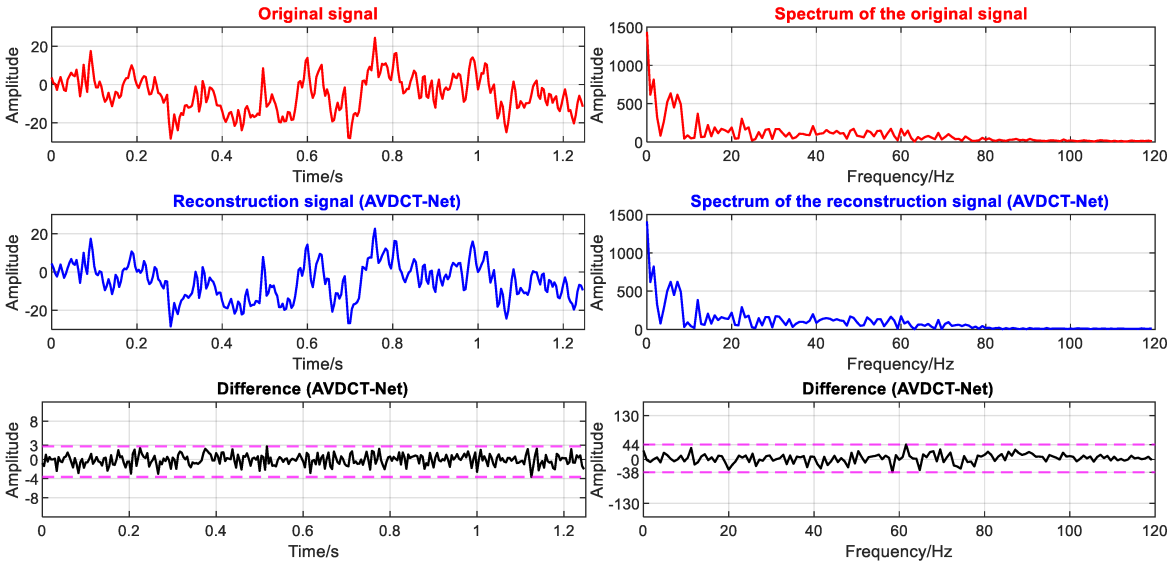
(b) Comparison between the original data and the reconstructed data on the BCI3 dataset.

Fig. 6: Comparison between the original data and the reconstructed data in the time domain (left) and the frequency domain (right) on the BCI2 dataset and BCI3 dataset.

subbands achieves a 3.44% improvement in CR, a 0.99% improvement in PRD, a 0.98% improvement in PRDN, and a 4.55% improvement in QS. The reason is that three scaling subbands in the DCT domain can be treated as three convolution layers in the time domain. Their coordinated operation synthesizes a filter bank that improves feature extraction efficiency. Furthermore, compared with other multiple subband models, the three-subband model exhibits the highest CR along with the lowest PRD and PRDN. Therefore, we select a subband quantity of three.

H. Computational Complexity and Compression Time

Table IV provides a comparative analysis of the trainable parameters and Multiply-Accumulate Operations (MACs) within the encoder modules across various algorithms. We focus on analyzing the encoder module because it is deployed in a resource-constrained edge gateway, whereas the decoder can be implemented on a more capable fog gateway. In this study, we leverage a 64-channel EEG signal as input, with each channel comprising 64 data samples captured over a duration of 0.27 seconds. Compared to the VAE, AVDCT-Net reduces the number of trainable parameters from 11,700 to 4,547, representing a 61.14% reduction. This significantly lowers the hardware memory requirements for parameter storage. Furthermore, AVDCT-Net decreases MACs from 737,280 to 561,152, a reduction of 23.89%, indicating a lower computational complexity. Additionally, AVDCT-Net has a comparable number of trainable parameters and MACs to ASAEDCT.

To evaluate the practicality of the proposed compression model in real-world scenarios, we analyze its actual energy

TABLE IV: The comparison of the number of trainable parameters and MACs.

Algorithm	Input size	Trainable parameters	MACs
ASAEDCT [11]	64×64	4,547	561,152
KCHE [9]	64×64	0	12,352
DCT-Perceptron [36]	64×64	129	274,432
VAE [37]	64×64	11,700	737,280
AVDCT-Net	64×64	4,547	561,152

TABLE V: The comparison of the power cost and runtime on Raspberry Pi 400.

Algorithm	Input size	Power cost (Joule)	Runtime (s)
ASAEDCT [11]	64×64	0.0510	0.0081
KCHE [9]	64×64	0.2904	0.0484
DCT-Perceptron [36]	64×64	0.0347	0.0055
VAE [37]	64×64	0.0617	0.0098
AVDCT-Net	64×64	0.0422	0.0067

consumption and inference latency on edge devices. For this study, the Raspberry Pi 400 [41], [42] is chosen as the edge computing platform. The Raspberry Pi 400 features a Broadcom BCM2711 system-on-chip (SoC) with a quad-core Cortex-A72 (ARM v8) 64-bit processor running at 1.8GHz. It is equipped with 4GB of LPDDR4-3200 SDRAM, enabling efficient multitasking and computing capabilities. Additionally, the primary energy consumption on the Raspberry Pi 400 occurs during the data compression process. It is computed as $E = P \times t$, where P is the power in watts (W), measured directly using a power meter, and t is the inference time in seconds (s). As shown in Table V, AVDCT-Net demonstrates a lower inference latency (0.0067s) and power cost (0.0422W) compared to ASAEDCT and VAE. The reason is that ASAEDCT employs an additional Tanh activation function compared to AVDCT-Net, while VAE utilizes more linear layers than AVDCT-Net. Although KCHE requires no trainable parameters and has fewer MACs per iteration, it exhibits higher inference latency and power consumption than AVDCT-Net. This is because KCHE demands numerous iterations to achieve convergence. In contrast, AVDCT-Net requires only a single forward pass during the inference stage. Furthermore, AVDCT-Net consumes 21.61% more energy and incurs 21.82% higher latency than DCTP. However, it improves compression quality by 35.29%, as shown in Table I. This trade-off highlights AVDCT-Net's ability to balance compression performance and power efficiency, making it well-suited for deployment on edge devices.

I. P300 Detection Experiment

The EEG data compression model is developed for remote BCI systems. Therefore, the P300 detection experiment is conducted to validate the effectiveness of the compression model. In the P300 detection experiment [20], the participants are shown a 6×6 matrix composed of 36 individual
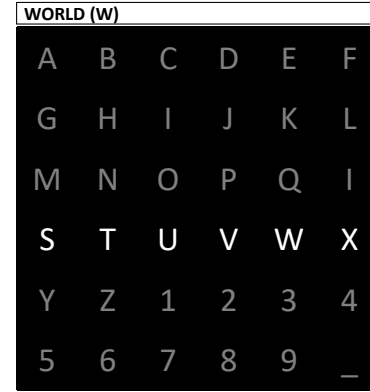

Fig. 7: A 6x6 P300 speller. In this example, the user is required to spell the word "WORLD" (one character at a time). For each character, the application randomly intensifies columns and rows several times (e.g., the fourth row in this instance)

alphanumeric characters as shown in Fig. 7. The task of the participants is to sequentially focus their gaze on lighted characters that are predetermined by the researcher. Each row and column of the 6×6 matrix was randomly intensified, yielding 12 distinct stimuli—6 corresponding to the rows and 6 to the columns. Among these intensifications, two specifically highlighted the target character. In this process, the brain activities of participants evoked by two stimuli with the target character differ from those evoked by the stimuli without the target character. Therefore, according to such differences, the BCI system can predict the character that participants were focusing on. The detailed experimental content is described in [20]. In this section, we utilize the BCI3 dataset because it provides a comprehensive record of P300 evoked potentials captured using the BCI2000 software. The paradigm utilized for these recordings is outlined in [13]. In the BCI3 datasets, the training sets contain recordings of 85 distinct characters, while the testing sets include recordings of 100 different characters. Each character is labeled as either a target or non-target. In addition, a single channel records 7,560 data points for each character.

The experiment begins with the application of the pre-trained AVDCT-Net encoder to compress the EEG data at the edge node as shown in Fig. 8. The compressed data is then converted into bitstreams for wireless transmission. Subsequently, the pre-trained AVDCT-Net decoder is utilized to restore the original EEG data at the fog node. After that, a spatial-temporal neural network (STNN) [43] is employed for P300 detection. The STNN has demonstrated superior performance across diverse P300 detection tasks. It is composed of a temporal unit and a spatial unit. The detailed structure of the STNN is illustrated in the supplementary material. Given that the authors in [43] also employed STNN for P300 detection on the BCI3 dataset, we adhere to the same data preprocessing, training, and testing procedures as described in their research. Specifically, the original recordings ATr and BTr are utilized as the training datasets, while the recordings ATe and BTe, reconstructed through compression models, are employed as the testing datasets.
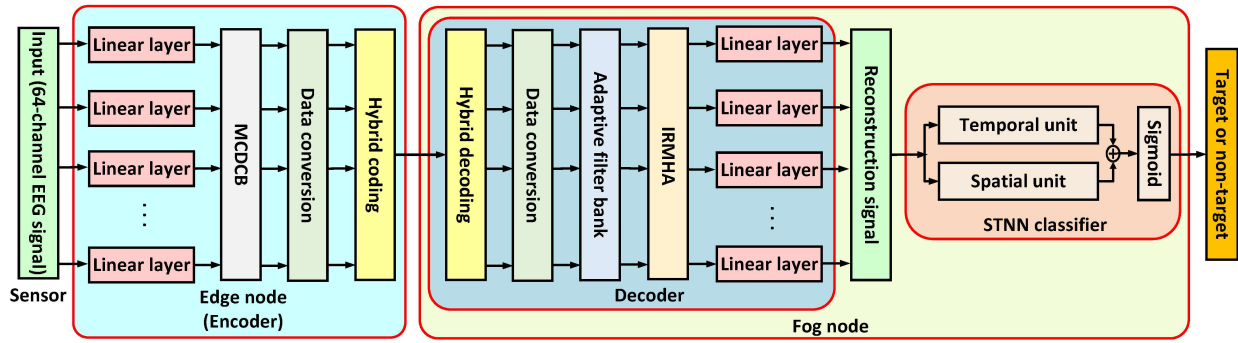
Fig. 8: P300 detection model consisting of the data compression/decompression and STNN network.

TABLE VI: P300 detection experiment on the BCI3 dataset.

Algorithm	Accuracy		
	Subject-A	Subject-B	Ave
Original+STNN	87.50%	90.92%	89.21%
ASAEDCT+STNN	87.58%	90.83%	89.21%
KCHE+STNN	87.00%	90.58%	88.79%
DCT-Perceptron+STNN	87.00%	91.00%	89.00%
VAE+STNN	86.92%	90.92%	88.92%
AVDCT-Net+STNN	**87.58%**	**91.17%**	**89.38%**

Second, the scaling layer works as a trainable filter to further emphasize important features. As shown in Fig. 6, AVDCT-Net effectively suppresses redundancy by discarding small coefficients in the frequency domain while preserving significant ones. Additionally, STNN classifies EEG signals by leveraging both temporal and global features. Meanwhile, the IRMHA mechanism in AVDCT-Net effectively captures a wide range of local and global dependencies from the feature map, thereby amplifying classification features.

As shown in Table VI, the STNN classifier achieved higher accuracy on datasets reconstructed by AVDCT-Net compared to the original datasets. It indicates that AVDCT-Net effectively preserves key classfication features during data compression and reconstruction. Additionally, AVDCT-Net is capable of eliminating redundant high-frequency components, such as noise, from EEG signals, which contributes to the improvement in accuracy. Furthermore, among datasets reconstructed by various compression models, the STNN classifier attains the highest average accuracy rate of 89.38% on those reconstructed by AVDCT-Net. This demonstrates that AVDCT-Net retains more important information than other models during the compression process. During compression, AVDCT-Net employs two key steps to preserve and enhance critical classification features in the DCT domain. First, a hard-thresholding layer filters out small DCT coefficients, which mainly represent noise and redundant details, while retaining large coefficients that hold vital classification information.

Fig. 9 shows the impact of different CR on the compression performance of AVDCT-Net and the accuracy rate of the STNN classifier, evaluated on dataset ATe and dataset BTe. In this experiment, the adjustment of CR is achieved by modifying the values of τ and ω as defined in Eq. (27). As the average CR increases from 11.68, both PRD and PRDN gradually rise, while the average accuracy of the STNN classifier on datasets reconstructed by AVDCT-Net gradually declines. This decline occurs because, with higher CR, more important information is lost from the signal. Additionally, it is observed that when the average CR is 14.89, the classifier's average accuracy aligns with its performance on original datasets (baseline). This implies that the average CR can be increased to at least 14.89 without compromising average classification accuracy. Moreover, AAEVDCT attains its highest QS at a CR of 14.89, indicating an optimal tradeoff between compression efficiency and reconstruction accuracy.
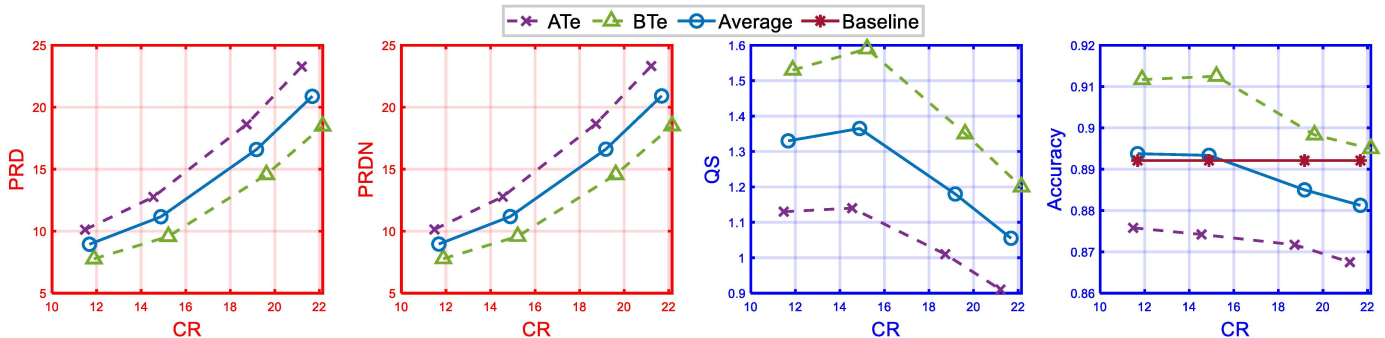

Fig. 9: Compression performance of AVDCT-Net and classification accuracy of STNN for different CRs.

V. CONCLUSION

In this article, we proposed a novel AVDCT-Net model for EEG data compression within an edge-fog computing architecture. The model leveraged a multi-channel structure to enable the concurrent compression of EEG signals across all channels, reducing both compression time and transmission latency. Specifically, a low-complexity encoder was deployed at the edge level. By imposing parallel hard-thresholding nonlinearities and scaling operators (filters) after the DCT, the redundant coefficients were removed, thereby enhancing the encoder's data compression capabilities. Besides, a new ELBO was derived to regularize the encoder output, promoting a more structured and efficient representation. At the fog level, an adaptive filter bank was adopted to merge relevant features from adjacent channels into each separate channel, improving the robustness and generalization ability. Furthermore, the IRMHA captured both local and global dependencies to reconstruct the original signal. The proposed model exhibited superior compression efficiency and reconstruction accuracy compared to state-of-the-art methods on the BCI2 and BCI3 datasets. Furthermore, as validated by the P300 detection experiment, AVDCT-Net preserved all critical features during the compression and reconstruction process. Therefore, the AVDCT-Net model was well-suited for implementation in an edge-fog computing framework for EEG data compression. In future work, AVDCT-Net will be deployed across a wider range of edge devices, including low-power IoT devices (e.g., Raspberry Pi, Arduino), embedded systems (e.g., FPGAs, DSPs), mobile devices (e.g., smartphones, tablets), and industrial/automotive edge computing platforms (e.g., NVIDIA Jetson, Qualcomm Snapdragon).

REFERENCES

[1] Khalida Douibi, Solène Le Bars, Alice Lemontey, Lipsa Nag, Rodrigo Balp, and Gabrièle Breda, "Toward eeg-based bci applications for industry 4.0: Challenges and possible applications," *Frontiers in Human Neuroscience*, vol. 15, pp. 705064, 2021.

[2] Baraka Maiseli, Abdi T Abdalla, Libe V Massawe, Mercy Mbise, Khadija Mkocha, Nassor Ally Nassor, Moses Ismail, James Michael, and Samwel Kimambo, "Brain–computer interface: trend, challenges, and threats," *Brain informatics*, vol. 10, no. 1, pp. 20, 2023.

[3] Yang An, Johnny Wong, and Sai Ho Ling, "Development of real-time brain-computer interface control system for robot," *Applied Soft Computing*, vol. 159, pp. 111648, 2024.

[4] Mahsa Bagheri and Sarah D Power, "Simultaneous classification of both mental workload and stress level suitable for an online passive brain–computer interface," *Sensors*, vol. 22, no. 2, pp. 535, 2022.

[5] Haroon Khan, Noman Naseer, Anis Yazidi, Per Kristian Eide, Hafiz Wajahat Hassan, and Peyman Mirtaheri, "Analysis of human gait using hybrid eeg-fnirs-based bci system: a review," *Frontiers in Human Neuroscience*, vol. 14, pp. 613254, 2021.

[6] Hamdi Altaheri, Ghulam Muhammad, and Mansour Alsulaiman, "Physics-informed attention temporal convolutional network for eeg-based motor imagery classification," *IEEE transactions on industrial informatics*, vol. 19, no. 2, pp. 2249–2258, 2022.

[7] Paula Ivone Rodríguez-Azar, Jose Manuel Mejía-Muñoz, Oliverio Cruz-Mejía, Rafael Torres-Escobar, and Lucero Verónica Ruelas López, "Fog computing for control of cyber-physical systems in industry using bci," *Sensors*, vol. 24, no. 1, pp. 149, 2023.

[8] Szczepan Paszkiel and Szczepan Paszkiel, "Using bci in iot implementation," *Analysis and Classification of EEG Signals for Brain–Computer Interfaces*, pp. 111–128, 2020.

[9] Ali Kadhum Idrees, Sara Kadhum Idrees, Raphael Couturier, and Tara Ali-Yahiya, "An edge-fog computing-enabled lossless eeg data compression with epileptic seizure detection in iomt networks," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13327–13337, 2022.

[10] V Hurbungs, V Bassoo, and TP Fowdur, "Fog and edge computing: concepts, tools and focus areas," *International Journal of Information Technology*, vol. 13, no. 2, pp. 511–522, 2021.

[11] Xin Zhu, Hongyi Pan, Shuaiang Rong, and Ahmet Enis Cetin, "Electroencephalogram sensor data compression using an asymmetrical sparse autoencoder with a discrete cosine transform layer," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 2160–2164.

[12] Darius Birvinskas, Vacius Jusas, Ignas Martisius, and Robertas Damasevicius, "Fast dct algorithms for eeg data compression in embedded systems," *Computer Science and Information Systems*, vol. 12, no. 1, pp. 49–62, 2015.

[13] Binh Nguyen, Wanli Ma, and Dat Tran, "A study of combined lossy compression and seizure detection on epileptic EEG signals," *Procedia Computer Science*, vol. 126, pp. 156–165, 2018.

[14] Behzad Hejrati, Abdolhossein Fathi, and Fardin Abdali-Mohammadi, "Efficient lossless multi-channel eeg compression based on channel clustering," *Biomedical Signal Processing and Control*, vol. 31, pp. 295–300, 2017.

[15] Abeer Z Al-Marridi, Amr Mohamed, and Aiman Erbad, "Convolutional autoencoder approach for EEG compression and reconstruction in m-health systems," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 370–375.

[16] Alberto Zancanaro, Giulia Cisotto, Italo Zoppis, and Sara Lucia Manzoni, "veegnet: learning latent representations to reconstruct eeg raw data via variational autoencoders," in *International Conference on Information and Communication Technologies for Ageing Well and e-Health*. Springer, 2023, pp. 114–129.

[17] Hugo Lerogeron, Romain Picot-Clémente, Laurent Heutte, and Alain Rakotomamonjy, "Learning an autoencoder to compress EEG signals via a neural network based approximation of DTW," *Procedia Computer Science*, vol. 222, pp. 448–457, 2023.

[18] Behzad Hejrati, Abdolhossein Fathi, and Fardin Abdali-Mohammadi, "A new near-lossless EEG compression method using ANN-based reconstruction technique," *Computers in Biology and Medicine*, vol. 87, pp. 87–94, 2017.

[19] Benjamin Blankertz, K-R Muller, Gabriel Curio, Theresa M Vaughan, Gerwin Schalk, Jonathan R Wolpaw, Alois Schlogl, Christa Neuper, Gert Pfurtscheller, Thilo Hinterberger, et al., "The bci competition 2003: progress and perspectives in detection and discrimination of eeg single trials," *IEEE transactions on biomedical engineering*, vol. 51, no. 6, pp. 1044–1051, 2004.

[20] Benjamin Blankertz, K-R Muller, Dean J Krusienski, Gerwin Schalk, Jonathan R Wolpaw, Alois Schlogl, Gert Pfurtscheller, Jd R Millan, Michael Schroder, and Niels Birbaumer, "The bci competition iii: Validating alternative approaches to actual bci problems," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 14, no. 2, pp. 153–159, 2006.

[21] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[22] Gregory K Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[23] Didier Le Gall, "Mpeg: A video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, no. 4, pp. 46–58, 1991.

[24] Lahiru D Chamain, Siyu Qi, and Zhi Ding, "End-to-end image classification and compression with variational autoencoders," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21916–21931, 2022.

[25] Murat Arican and Kemal Polat, "Pairwise and variance based signal compression algorithm (pvbsc) in the p300 based speller systems using eeg signals," *Computer methods and programs in biomedicine*, vol. 176, pp. 149–157, 2019.

[26] David L Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[27] Stephen A Martucci, "Symmetric convolution and the discrete sine and cosine transforms," *IEEE Transactions on Signal Processing*, vol. 42, no. 5, pp. 1038–1051, 1994.

[28] HyunWook Park, YoungSeo Park, and Seung-Kyun Oh, "L/M-fold image resizing in block-DCT domain using symmetric convolution," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1016–1034, 2003.

[29] Justin Dauwels, K Srinivasan, M Ramasubba Reddy, and Andrzej Cichocki, "Near-lossless multichannel eeg compression based on matrix and tensor decompositions," *IEEE journal of biomedical and health informatics*, vol. 17, no. 3, pp. 708–714, 2012.

[30] Xin Zhu, Daoguang Yang, Hongyi Pan, Hamid Reza Karimi, Didem Ozevin, and Ahmet Enis Cetin, "A novel asymmetrical autoencoder with a sparsifying discrete cosine stockwell transform layer for gearbox sensor data compression," *Engineering Applications of Artificial Intelligence*, vol. 127, pp. 107322, 2024.

[31] Andrew Ng et al., "Sparse autoencoder," *CS294A Lecture Notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[32] Edmund Y Lam and Joseph W Goodman, "A mathematical analysis of the dct coefficient distributions for images," *IEEE transactions on image processing*, vol. 9, no. 10, pp. 1661–1666, 2000.

[33] Shahin Akhter and MA Haque, "ECG compression using run length encoding," in *2010 18th European Signal Processing Conference*. IEEE, 2010, pp. 1645–1649.

[34] Zongjie Tu and Shiyong Zhang, "A novel implementation of JPEG 2000 lossless coding based on LZMA," in *The Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*. IEEE, 2006, pp. 140–140.

[35] Paruj Ratanaworabhan, Jian Ke, and Martin Burtscher, "Fast lossless compression of scientific floating-point data," in *Data Compression Conference (DCC'06)*. IEEE, 2006, pp. 133–142.

[36] Hongyi Pan, Emadeldeen Hamdan, Xin Zhu, Salih Atici, and Ahmet Enis Cetin, "Multichannel orthogonal transform-based perceptron layers for efficient resnets," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[37] Adaiton Oliveira-Filho, Ryad Zemouri, Philippe Cambron, and Antoine Tahan, "Early detection and diagnosis of wind turbine abnormal conditions using an interpretable supervised variational autoencoder model," *Energies*, vol. 16, no. 12, pp. 4544, 2023.

[38] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[39] Filippos Christianos, Georgios Papoudakis, Muhammad A Rahman, and Stefano V Albrecht, "Scaling multi-agent reinforcement learning with selective parameter sharing," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1989–1998.

[40] Baosu Guo, Zhaohui Qiao, Hao Dong, Zhen Wang, Shuiquan Huang, Zhengkai Xu, Fenghe Wu, Chuanzhen Huang, and Qing Ni, "Temporal convolutional approach with residual multi-head attention mechanism for remaining useful life of manufacturing tools," *Engineering Applications of Artificial Intelligence*, vol. 128, pp. 107538, 2024.

[41] Raspberry Pi Foundation, "Raspberry Pi 400 Product Brief," 2022, Accessed: Feb. 16, 2025.

[42] Ahmad Saeed Mohammad, Thoalfeqar G Jarullah, Musab TS Al-Kaltakchi, Jabir Alshehabi Al-Ani, and Somdip Dey, "Iot-mfacenet: Internet-of-things-based face recognition using mobilenetv2 and facenet deep-learning implementations on a raspberry pi-400," *Journal of Low Power Electronics and Applications*, vol. 14, no. 3, pp. 46, 2024.

[43] Zhen Zhang, Xiaoyan Yu, Xianwei Rong, and Makoto Iwata, "Spatial-temporal neural network for p300 detection," *IEEE Access*, vol. 9, pp. 163441–163455, 2021.

H. Pan received his B.S. degree in Automation from Chang'an University, Xi'an, China, in 2018, and his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Illinois Chicago, Chicago, IL, USA, in 2019 and 2023, respectively. He currently works as a Postdoctoral Research Fellow in the Machine and Hybrid Intelligence Laboratory at Northwestern University, Chicago. His research interests include signal processing, machine learning, and biomedical image analysis. He serves as an Associate Editor of Signal, Image and Video Processing.

A. E. Cetin received his B.Sc. from METU, Ankara, Turkey, and Ph.D. in 1987 from the University of Pennsylvania, USA. He was an Assistant Prof. at the University of Toronto between 1987-1989 in Canada. He was a faculty member at Bilkent University from 1987 to 2017. He is currently a professor in the Department of Electrical and Computer Engineering at the University of Illinois Chicago (UIC). He also held visiting professor positions at Bellcore (1988), University of Minnesota (1996-1997), and UC San Diego (2016-2017). He has been carrying out research in the areas of theoretical and applied machine learning, signal, image, and video processing, biomedical signal processing, infrared and chemical sensor signal processing in Cyber-Physical Systems (CPS). His group introduced the concept of adaptive prediction and split vector quantization for Line Spectral Frequency representation. This concept was used in ITU speech coding standards including G.729, G.723.1, and GSM EFR.

He became a Fellow of IEEE for his contributions to signal and image recovery. He is the Editor-in-Chief of Signal, Image and Video Processing, Springer-Nature. He received a best paper award for his camera-based wildfire detection work at a conference organized by UNESCO and Cyprus Presidency of the European Union. He is one of the co-founders of the multinational smart wide-angle OEM camera company Oncam-Grandeye (https://www.oncamgrandeye.com), UK. He served as the CEO/CTO of Grandeye, Turkey between 2003-2013. Oncam-Grandeye cameras won design and innovation awards in IFSEC, UK, and ISC WEST, Las Vegas, trade fairs.

X. Zhu received the M.S. degree from the School of Telecommunications Engineering, Xidian University, Xi'an, China, in 2022. He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Illinois Chicago, Illinois, USA. His research interests include signal processing, machine learning, data compression, random access, and biomedical image analysis.