Comparative Analysis of Traditional, Hypercomplex, and Pre-trained Deep Neural Networks for Audio Emotion Recognition

1st Omar Ghezzi
University of Milan
omar.ghezzi@studenti.unimi.it

***Abstract*—This paper presents a comparative study of traditional, hypercomplex, and pre-trained deep neural networks for the Audio Emotion Recognition (AER) task. We introduce novel hypercomplex models, including `CliffSER1D`, `CliffSER2D`, `PureCliffSER1D` and `PureCliffSER2D`, designed to predict discrete emotional classes from hand-crafted MFCCs and log-Mel spectrogram features in both 1D and 2D domains. Our pipeline includes experiments with Empirical Mode Decomposition (EMD) pre-processing analysis. Additionally, we propose `CliffW2V`, a hypercomplex projection-head architecture fine-tuned for AER tasks, built on the end-to-end `Wav2Vec` pre-trained transformer model. Results show that the proposed `CliffSER1D`, `CliffSER2D` models outperform traditional approaches, when leveraging scalar 1D and 2D hand-crafted logMel features, without EMD preprocessing.**

***Index Terms*—Audio Emotion Recognition, Hypercomplex Neural Networks, Clifford Algebra, Empirical Mode decomposition**

I. Introduction

Emotions are intricate psychological states consisting of personal experiences, physiological responses, behavioral reactions, and communicative expressions [1], [2]. Affective Computing (AC) is the research domain dedicated to recognizing, understanding, simulating, and eliciting emotions within computational systems [3]. Among its tasks, emotion recognition is extensively explored in AC literature, involving computational models that link observed multimedia signals to specific emotional manifestations. Emotion detection holds broad practical utility, such as in crime investigation, psychiatric diagnosis, fatigue detection, disease diagnosis support, biometrics, and human-computer interaction. Despite significant research efforts, emotion detection remains challenging due to the subjective nature of emotions, the complexity of extracting meaningful descriptors from raw signals, and the difficulty in selecting optimal computational models to approximate the data-emotion relationship [1], [6].

In the psychological study of emotions, two primary theories have emerged: the discrete emotional model and the dimensional emotional model. The discrete model, proposed by Ekman [4], categorizes emotions into six fundamental types: sadness, happiness, fear, anger, disgust, and surprise. These emotions are considered innate and culturally universal. Conversely, the dimensional model, exemplified by Russell's circumplex model of affect [5], characterizes emotions along a few latent dimensions, suggesting that emotions arise from two independent neural systems: valence (pleasantness or unpleasantness) and arousal (intensity). While the dimensional model offers a more nuanced understanding, it is less intuitive and harder to label accurately. Additionally, some basic emotions can overlap or fall outside the defined dimensions.

This study investigates Audio Emotion Recognition (AER), a subfield of Affective Computing focused on analyzing emotions in audio signals. AER employs audio features to highlight intrinsic signal characteristics correlated with recorded emotional expressions. These features are typically classified into prosodic, spectral, and voice quality categories [7]. Prosodic features, such as rhythm, intonation, energy, duration, and fundamental frequency, are based on human perception. Spectral features, including Mel-frequency cepstral coefficients (MFCCs) and linear prediction coefficients (LPCs), capture vocal cord properties. Voice quality features, like jitter, harmonics-to-noise ratio, and shimmer, explore the relationship between vocal tract characteristics and emotional content.

To model the intrinsic relationship between audio signals and their corresponding emotional categorization, this project utilizes Hypercomplex Deep Learning (HDL) architectures [8], [9]. Traditional real-valued deep learning models represent features in an n-dimensional real vector space, using "feature mapping" to non-linearly transform each input into a higher-dimensional space. This process aligns a learnable non-linear function in the original space with a separating hyperplane in the expanded space. Deep networks achieve this through sequential transformations across multiple finite-dimensional hidden layers.

However, these networks often overlook the physical or geometric inter- and intra-feature dependencies, such as vector fields or complex numbers. Additionally, they rely solely on the inner product operator (dot product plus a non-linear activation function) in each intermediate layer to determine the next output vector component [10]. In contrast, in the hypercomplex framework, data is modeled using hypercomplex numbers, or multivectors, as elements of a Clifford algebra of specific signature. The learned hypothesis in this framework

is constructed using the Clifford product, which provides clear geometric significance and better captures the inherent relationships in the data [11].

Relying on the discrete emotional model, this project performs the emotion classification task on spectral handcrafted features (MFCCs and the entire log-Mel spectrogram of the audio signal) exploiting four HDL models, `CliffSER1D`, `PureCliffSER1D`, `CliffSER2D` and `PureCliffSER2D`. In light of the findings presented in [16], to support the hypercomplex data representation, emphasizing intra-channel correlations in multimodal signals, and to handle the non-stationary characteristics of speech signals, we also use empirical mode decomposition (EMD).

II. Theoretical Background

A. Speech Signal Processing

Speech can be modeled as the output $y[t]$ of a linear-and-time-varying system $V[\cdot]$, or $U[\cdot]$, whose properties vary slow with time, applied to a quasi-period impulse train $\delta_P[t]$ or a random noise signal $\mu[t]$ [12]. It is therefore possible to model *voiced speech* signals as follows:

$$y[t] = V[\delta_P[t]] = \delta_P[t] * h_V[t] \quad , \tag{1}$$

with $h_V[t]$ being the impulse response of $V[\cdot]$:

$$h_V[t] = A_V \left(g[t] * v[t] * r[t]\right) \quad , \tag{2}$$

where $g[t]$ is the glottal pulse signal, $v[t]$ the vocal tract impulse response, $r[t]$ the radiation load response at the lips and A_V the voice gain.

As we can see, convolution is at the heart of the discrete-time models for speech production. Cepstral analysis is a way to easily separate the components of a signal obtained as a result of a multiple convolution. By exploiting the logarithm product rule, convolutions in the time domain (equivalent to products in the frequency domain) are transformed into sums in the "quefrency" domain. This approach simplifies the separation of vocal-related components from noise sources.

Mel-frequency cepstral coefficients (MFCCs) are the most widely used method for cepstral analysis. MFCCs are derived by taking the first M points of the cepstral transform, using Discrete Cosine Transform (DCT), of R sub-bands extracted from the logMel-spectrogram of the τ-th frame of an input signal. Formally, $\forall m = 1, \ldots, M$:

$$\text{MFFC}_\tau[m] = \frac{1}{R}\sum_{r=1}^{R} \log\big(\tilde{X}_\tau[r]\big) \cos\Big(\frac{2\pi}{R}\big(r+\frac{1}{2}\big)m\Big) \quad , \tag{3}$$

were the R logMel-spectrum sub-bands are obtained as follows, $\forall r = 1, \ldots, R$:

$$\tilde{X}_\tau[r] = \frac{1}{\sum_{k=0}^{N-1} |\text{Tri}_r[k]|^2} \sum_{k=0}^{N-1} \left|\text{Tri}_r[k] X_\tau[k]\right|^2 \quad . \tag{4}$$

Here, $\{\text{Tri}_r[k]\}_{r=1}^{R}$ is a family of R triangular weighting functions defined in the discrete frequency domain.

B. Clifford Algebra

The Clifford algebra [13], [14] is a specific algebraic structure, known as algebra over a field, which unifies heterogeneous substructures into a single environment. This algebra blurs the distinction between vector spaces and numerical fields, allowing the representation and multiplication of elements belonging to both domains. Formally, an algebra V over a Field K is a vector space $(V, +_V, \cdot_{KV})$, implicitly defined on the field $(K, +_K, \cdot_K)$ of scalars, together with a bilinear composition law $\cdot_V : V \times V \to V$. The Clifford Algebra over the real field $\mathbb{R}$ enriches $\mathbb{R}^n$ with a specific bilinear operator known as the Clifford product.

Definition 2.1 (Real Clifford algebra): Consider the vector space $(\mathbb{R}^n, +_{\mathbb{R}^n}, \cdot_{\mathbb{R}\mathbb{R}^n})$[1] over the field $(\mathbb{R}, +_{\mathbb{R}}, \cdot_{\mathbb{R}})$, equipped with a non-degenerate quadratic form $q_A(\cdot) : \mathbb{R}^n \to \mathbb{R}$ such that the corresponding diagonalized and normalized form $q_G(\cdot)$ satisfies:

$$q_G(\boldsymbol{e}_i) = \begin{cases} 1 & 1 \le i \le p \\ -1 & p < i \le p+q \end{cases} \tag{5}$$

The Clifford algebra $(Cl_{p,q}, +_{Clp,q}, \cdot_{Clp,q})$ is the real associative algebra constructed on $\mathbb{R}^n$ by requiring that for each canonical vector $\boldsymbol{e}_1, ..., \boldsymbol{e}_n \in \mathbb{R}^n$ and for each of its representatives $e_1, ..., e_n \in Clp, q$, the product $\cdot_{Cl_{p,q}}$ between elements of the algebra satisfies the following conditions:

$$e_i^2 = q_A(\boldsymbol{e}_i), \quad \forall i = 1, ..., p+q \tag{6}$$

$$e_i e_j = -e_j e_i \quad \forall i < j \tag{7}$$

The Clifford product $\cdot_{Cl_{p,q}}$ can be equivalently represented in matrix form. Specifically, for any pair of elements in a four-dimensional Clifford algebra $x, \omega \in Cl_{p,q}$, $2^{p+q} = 4$, the coefficients of the "right" product $\omega \cdot_{Cl_{p,q}} x$, between ω and x, take on the following matrix form[2]:

$$W_\omega^R \, \underline{x} = \begin{pmatrix} \omega_0 & \gamma_1\omega_1 & \gamma_2\omega_2 & -\gamma_1\gamma_2\omega_{12} \\ \omega_1 & \omega_0 & -\gamma_2\omega_{12} & \gamma_2\omega_2 \\ \omega_2 & \gamma_1\omega_{12} & \omega_0 & -\gamma_1\omega_1 \\ \omega_{12} & \omega_2 & -\omega_1 & \omega_0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_{12} \end{pmatrix} \quad , \tag{8}$$

The coefficients of the right product of two elements in an eight-dimensional Clifford algebra can be computed using similar matrix representations (for the detailed explanation, please refer to [**?**]).

C. Clifford neural layers

Before introducing the definition of Clifford linear layers [9], it is helpful to review classic linear layers. These layers

[1]More precisely, $\mathbb{R}^{p,q}$ represent a space where p denotes he euclidean component and and q denotes the anti-euclidean part. To illustrate, consider $\mathbb{C}$ as a real algebra on $\mathbb{R}^{0,1}$. Here, the anti-euclidean part corresponds to the imaginary component.

[2]To ensure consistent notation, we introduce three symbols for denoting the i-th basis element in $\mathbb{R}^n$. For instance, in $\mathbb{R}^2$ and $Cl_{2,0}$, $\mathbf{e}_1 = (1,0)^T$ represents a vector in $\mathbb{R}^2$, $e_1 = 0 + 1e_1 + 0e_2 + 0e_{12}$ represents an element of $Cl_{2,0}$, and $\underline{e}_1 = (0,1,0,0)^T$ denotes the coefficients of e_1, i.e a vector in $\mathbb{R}^{2^{p+q}}$, where $p = 2, q = 0$.

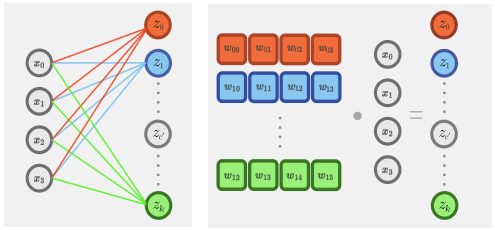

Fig. 1. Graphical description of a classical linear layer in a MLP model.

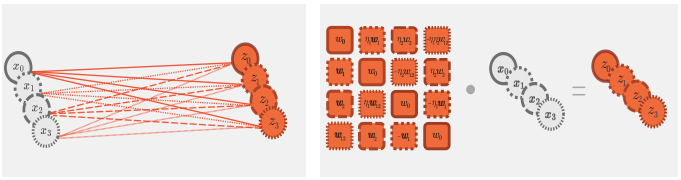

Fig. 2. Graphical description of a Clifford linear layer (in a four-dimensional Clifford algebra).

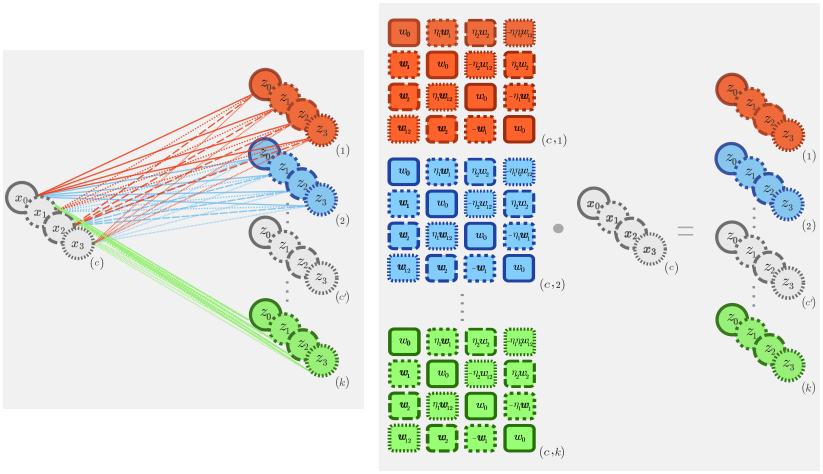

Fig. 3. Graphical description of a Clifford linear layer with with feature expansion and activation function (in a four-dimensional Clifford algebra).

are applications $l : \mathbb{R}^d \to \mathbb{R}^k$, followed by a component-wise non-linear activation function $\sigma : \mathbb{R} \to \mathbb{R}$, s.t:

$$z_{c'}^{(l)} = \sigma\left(\sum_{c=1}^{d} w_{c,c'}^{(l)} \cdot_{\mathbb{R}} x_c\right) \quad , \quad \forall c' = 1, ..., k \quad . \tag{9}$$

In other words, the output projection onto the c'-th direction, in $\mathbb{R}^k$, is determined by the inner product, in $\mathbb{R}^d$, between the c'-th learnable weight vector $\boldsymbol{w}_c \in \mathbb{R}^d$ and the input $\boldsymbol{x} \in \mathbb{R}^d$. This projection is then passed through the non-linear activation function σ. A graphical representation of a classical linear layer is depicted in fig. 1.

Definition 2.2 (Clifford linear layer): Clifford linear layer is a mapping $l : Cl_{p,q} \to Cl_{p,q}$ that operates by multiplying the input $x \in Cl_{p,q}$ by a learnable element $\omega^{(l)} \in Cl_{p,q}$ using the Clifford product.

$$\underline{z}^{(l)} = W_{\omega}^{R}\, \underline{x} \ . \tag{10}$$

Any Clifford linear layer learns a multivector $\omega^{(l)} \in Cl_{p,q}$ and maps the input $x \in Cl_{p,q}$ to the output $z \in Cl_{p,q}$ exploiting the matrix form of the right Clifford product in 8. Una rappresentazione grafica del layer lineare di Clifford è visualizzabile in fig. 2. The Clifford product used in 10 can linearize an orthogonal transformation of the input without needing feature mapping or non-linear functions. This capability is inherently tied to the *inductive bias* introduced by using Clifford algebra in the learning process.

Consider the Clifford algebra $Cl_{2,0}$, built on $\mathbb{R}^2$ with basis $\{1, e_1, e_2, e_{12}\}$. Since $e_{12}^2 = -1$, it contains a copy of $\mathbb{C}$ as a sub-structure:

$$\mathbb{C} \simeq \{x + 0e_1 + 0e_2 + ye_{12} | x + iy \in \mathbb{C}\} \subset Cl_{2,0} \tag{11}$$

Multiplying a complex number by $e^{i\theta} \in \mathbb{C}$ typically results in a counter-clockwise rotation by θ on the Gauss plane. To enable a standard neural network to learn such a rotation, it would be intuitive to represent both the input and the weights in $\mathbb{C}$, or better yet, represent the input as a vector in $\mathbb{R}^2$ and the learnable weight in $\mathbb{C}$. While this is not possible in standard neural network data space ($\mathbb{R}^n$), it is natural in $Cl_{2,0}$. We can represent the input as $x = x_1 e_1 + x_2 e_2$ and let the network learn $\omega^{(l)} = 1 \cos\theta + e_{12} \sin\theta$.

Despite the advantages mentioned, applying feature expansion and non-linear activation functions within the context of Clifford algebra can still benefit the model, particularly when working with single-channel input data (i.e., when the only non-zero coefficient of $\underline{x}$ corresponds to the scalar unit in the algebra's basis). This approach can also be advantageous if the previously described model suffers from high bias (tendency to underfit).

Definition 2.3 (Clifford linear layer with feature expansion and activation function): Clifford linear layer is a mapping $l : Cl_{p,q}^{d} \to Cl_{p,q}^{k}$ that operates by multiplying the input $x_c \in Cl_{p,q}$ by a learnable element $\omega^{(l)} \in Cl_{p,q}$ using the Clifford

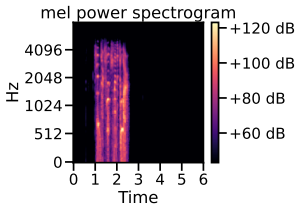

Fig. 4. High Frequencies IMFs.

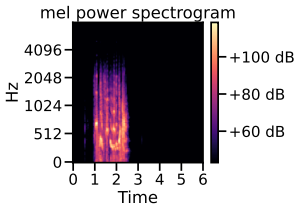

Fig. 5. Medium Frequencies IMFs.

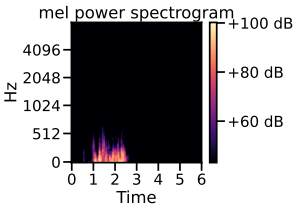

Fig. 6. Low Frequencies IMFs.

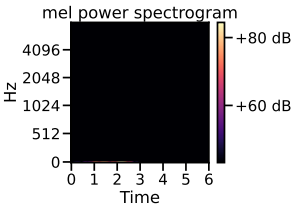

Fig. 7. Very Low Frequencies IMFs.

product.

$$\underline{z}_{c'}^{(l)} = \sigma\left(\sum_{c=1}^{d}(W_{\omega}^{R})_{c,c'}\,\underline{x}_{c}\right) \quad . \tag{12}$$

A graphical representation of the Clifford linear layer is shown in fig. 3.

It is important to note that the formulation in 10, along with its 1D and 2D convolutional analogs, will form the basis of the `PureCliffSER1D` and `PureCliffSER2D` models. These models are specifically designed to fully exploit the inductive bias provided by Clifford algebra. Conversely, the formulation in 12 will be a fundamental component of the `CliffSER1D` and `CliffSER2D` models, striking a balance between traditional layers and those deeply rooted in the algebra's semantics.

D. Empirical Mode decomposition

Empirical Mode Decomposition (EMD) is an analytical technique used to characterize non-linear and non-stationary time series by decomposing them into intrinsic mode functions (IMFs) [15]. Instead of representing dynamic signals with combinations of static basis functions, as the Fourier transform does, EMD looks to isolate a small number of temporally adaptive basis functions (the IMFs) and derive dynamics in frequency and amplitude directly from them [17]. Formally, an IMF is characterized by specific conditions:

1) Number of zero-crossing and extrema: the number of extrema (signal's maximum and minimum amplitudes) and the number of zero-crossing must be either equal or differ at most by 1.
2) Local-symmetry around zero: The function must be symmetric with respect to local zero mean.

IMFs are extracted from the data using the 'sift' algorithm, a time-domain method that does not rely on Fourier transforms to separate different source components.

Given an arbitrary, non-stationary and non-linear signal $x[t]$, the sift algorithm works iteratively by extracting its IMFs components, from the faster oscillation $c_1[t]$ to the very slowest one $c_n[t]$, with a residual non-stationary trend $r_n[t]$ left. The sift algorithm iterative steps are:

1) Extraction of the local maxima and local minima of $x[t]$.
2) Cubic spline interpolation $e_1^{max}[t]$ and $e_1^{min}[t]$, respectively of the maxima and the minima.
3) Computing the mean envelope:

$$e_1^{mean}[t] = \frac{e_1^{max}[t] + e_1^{min}[t]}{2}$$

4) The current IMF is initially extracted by removing the mean envelope from the previous data:

$$h_1[t] = x[t] - e_1^{mean}[t]$$

If $h_1[t]$ doesn't satisfy the IMFs' conditions, $x[t]$ is replaced by $h_1[t]$ in the previous 1) to 4) steps, leading to:

$$e_{1,1}^{mean}[t] = \frac{e_{1,1}^{max}[t] + e_{1,1}^{min}[t]}{2}$$

and:

$$h_{1,1}[t] = h_1[t] - e_{1,1}^{mean}[t]$$

If also $h_{1,1}[t]$ doesn't satisfy the IMFs' conditions 1 and , it is necessary to iteratively repeat k times the previous-data-replacing step, obtaining:

$$h_{1,k}[t] = h_{1,(k-1)}[t] - e_{1,k}^{mean}[t]$$

The IMFs' conditions are evaluated in terms of the standard deviation of two subsequent results, e.g. $h_{1,k}[t]$ and $h_{1,k+1}[t]$. If the following condition holds:

$$\sum_{t=1}^{T}\frac{(h_{1,(k-1)}[t] - h_{1,k}[t])^2}{h_{1,(k-1)}[t]^2} \leq 0.1$$

then $h_{1,k}[t]$ can be considered an IMF:

$$c_1[t] = h_{1,k}[t]$$

5) Separate the current IMF from the data

$$r_i[t] = x[t] - c_{i-1}[t]$$

and use the residual component to repeat steps 1) to 5), replacing $x[t]$. The algorithm stops when the n-th residual $r_n[t]$ results in a monotonic function or a function with less than two local maximum or minimum.

A visual representation of log-mel spectrograms for four EMD groups — High Frequency (HF), Medium Frequency (MD), Low Frequency (LF), and the residual signal — for an input speech audio signal is shown in fig. 4, 5, 6 and 7.

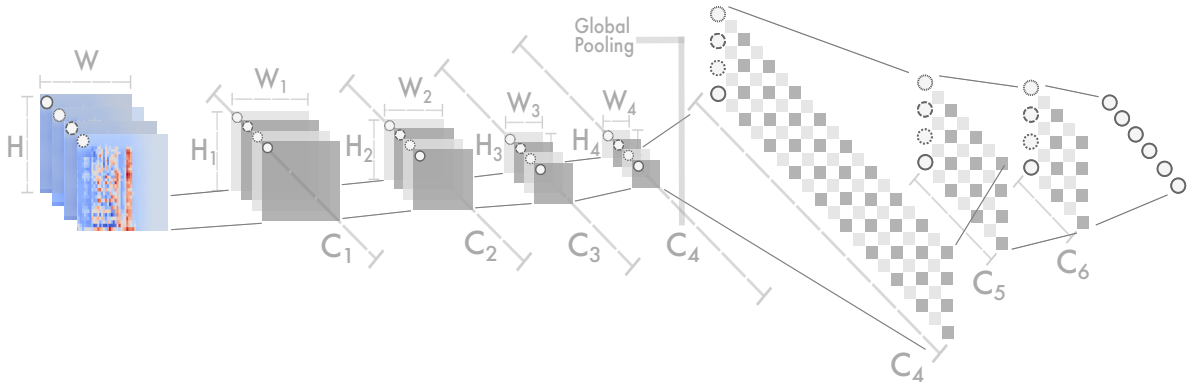

Fig. 8. CliffSER2D graphic model.

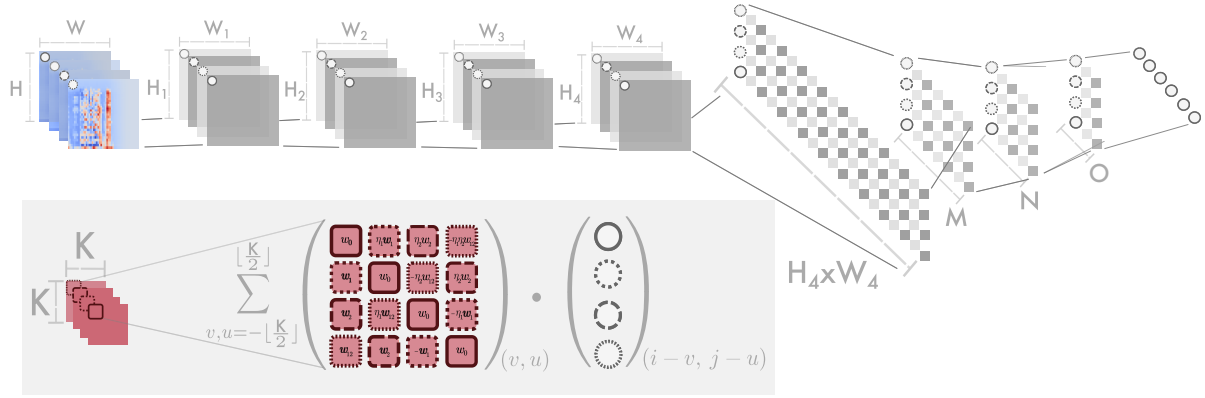

Fig. 9. PureCliffSER2D graphic model.

E. Representation learning through self-supervision

A good data representation simplifies the resolution of subsequent, more specific learning tasks [10]. Representation learning aims to achieve this by embedding large amounts of possibly unlabelled and unrefined data into a lower-dimensional, general-purpose latent space that is invariant to pre-selected noisy factors. This intermediate representation facilitates improved performance in downstream predictive tasks on smaller, usually labelled benchmarks. The process involves two steps: first, learning useful intermediate data representations through a *pretext* task, and second, utilizing these representations in a *downstream* task (i.e. traditional supervised regression or classification).

Representation learning seeks to find data representations that act as effective substrates for prediction. Self-supervision, which treats an unsupervised setting as a supervised task by creating pseudo targets/labels from the raw data itself, is an effective way to achieve good intermediate data representation. Self-supervision can be realized in either a generative or a contrastive manner [18].

Generative self-supervision is similar to the unsupervised setting involving autoencoders, but with a key difference: instead of simply reconstructing the original data, the model learns to complete missing parts of previously masked input data, treating the omitted portions as pseudo labels.

Contrastive self-supervision bake invariances to undesired data charecteristics into the objective function, penalizing deviations from the required constraints. It aligns similar datapoints (or tokens/patches of the same data) to similar embeddings, while pushing then apart from non-cooccurring embeddings. Simultaneously, it encourages latent representations to be uniformly and evenly distributed within a simple topological space, typically an n-dimensional hypersphere [19].

The *Wav2Vec* model [20], [21] combines both contrastive and generative self-supervised learning within the following theoretically formalized pretext problem:

$$\ell : \mathcal{X} \times \mathcal{F} \times \mathcal{G} \;\rightarrow\; [0, +\infty) \;: \tag{13}$$

$$\boldsymbol{x}, f, g, \mapsto\; \ell\Big((g \circ f)(\boldsymbol{x}),\; (q \circ f)(\boldsymbol{x}) \Big) \quad . \tag{14}$$

Here, $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_T = f(\boldsymbol{x})$ are the latent representations of T partially overlapping temporal windows of the original signal. The network $f : \boldsymbol{X} \rightarrow \boldsymbol{Z}$ is the transformer encoder backbone. The loss function aims to align the *anchor* $\boldsymbol{z}_i$ with its *attractor* element by matching the correct quantized latent pseudo-label $q(\boldsymbol{z}_i)$ with the output $g(\boldsymbol{z}_i)$of the context network decoder $g(\cdot)$. Simultaneously, it seeks to misalign $g(\boldsymbol{z}_i)$ with the *distractors* $q(\boldsymbol{z}_j)$, $j \neq i$. Pseudo labels are generated

through a quantization module $q : \boldsymbol{X} \rightarrow \boldsymbol{Q}$, which maps latent representations into a finite codebook representation.

III. Methods

A. Models

In this section, we introduce the `CliffSER` and `PureCliffSER` families of models. The `CliffSER` models utilize feature expansion and non-linear activation functions to enhance predictive capabilities within the hypercomplex framework, while mitigating the inherent bias introduced by Clifford algebra. On the other hand, `PureCliffSER` models adhere strictly to the algebraic structure, abstaining from predictions in higher-dimensional spaces and external non-linear transformations.

1) `CliffSER1D`: The `CliffSER1D` model processes a 128-dimensional input vector consisting of time-axis-averaged MFCCs or time-axis-averaged logMel spectrogram data. Each element of this input vector is treated as the scalar component of an element $x \in Cl_{p,q}$, with q, p as hyperparameters. The entire vector is denoted as $\boldsymbol{x} \in Cl^S_{p,q}$, with $S = 128$. The model proceeds by passing the input through four 1D-Clifford convolutional layers, collectively performing the following transformation:

$$l_{1-4} : Cl^S_{p,q} \rightarrow Cl^{N\times C_4}_{p,q}, \qquad \boldsymbol{x} \mapsto \boldsymbol{z}^{(l_4)} \quad , \tag{15}$$

where the kernel size K_{1D} and the stride S are hyperparameters. Subsequently, a global average pooling layer condenses the temporal dimension N to a single unit:

$$l_5 : Cl^{N\times C_4}_{p,q} \rightarrow Cl^{C_4}_{p,q}, \qquad \boldsymbol{z}^{(l_4)} \mapsto \boldsymbol{z}^{(l_5)} \quad . \tag{16}$$

The resulting output $\boldsymbol{z}^{(l_5)}$ is then fed into three Clifford Linear layers, structured as introduced in 12, collectively performing the transformation:

$$l_{6-8} : Cl^{C_4}_{p,q} \rightarrow Cl^{6}_{p,q}, \qquad \boldsymbol{z}^{(l_5)} \mapsto \boldsymbol{z}^{(l_8)} \quad . \tag{17}$$

Finally, a conventional linear layer combines the multivector components of $\boldsymbol{z}^{(l_8)}$, resulting in six scalars that denote the probability of the input belonging to each of the six emotion classes. All layers, except the output layer, utilize a leaky ReLU activation function. Dropout layers with a dropout rate of 0.3 follow each dense layer, contributing to regularization by randomly excluding a fraction of input units during training to prevent overfitting.

2) `CliffSER2D`: The `CliffSER2D` model processes a 128×188 input matrix, representing either the 128 MFCC bands or the 128 logMel spectrogram frequency points, both discretized in 188 short-term temporal windows. Each pixel is modeled as the scalar component of an algebra element $x \in Cl_{p,q}$, allowing the entire image to be denoted as $\boldsymbol{x} \in Cl^{H\times W}_{p,q}$, with $H = 128$, $W = 188$. The model comprises four 2D-Clifford convolutional neural layers, collectively performing the following mapping:

$$l_{1-4} : Cl^{H\times W}_{p,q} \rightarrow Cl^{H_4\times W_4\times C_4}_{p,q}, \qquad \boldsymbol{x} \mapsto \boldsymbol{z}^{(l_4)} \quad . \tag{18}$$

The signature q, p, the kernel size K_{2D} and the stride S are hyperparameters. Global average pooling reduces the output from $Cl^{H_4\times W_4\times C_4}_{p,q}$ to $Cl^{C_4}_{p,q}$. Three Clifford Linear layers performs the same mapping as in 17, while the final traditional layer leads to the predicted six scalars. The `CliffSER2D` model uses leaky ReLU activations and dropout layers after each convolutional and linear Clifford layer to enhance regularization. A graphical representation of the architecture is depicted in fig. 8.

3) `PureCliffSER1D`: The `PureCliffSER1D` processes a 128-dimensional vector of time-averaged MFCC coefficients or the time-averaged full logMel spectrum. It employs four 1D convolutional layers without feature expansion techniques or non-linear activation functions, simply filtering the input data and possibly reducing the data length from $S = 128$ to N:

$$l_{1-4} : Cl^S_{p,q} \rightarrow Cl^{N}_{p,q}, \qquad \boldsymbol{x} \mapsto \boldsymbol{z}^{(l_4)} \quad . \tag{19}$$

4) `PureCliffSER2D`: The `PureCliffSER2D` model processes the same 128×188 input matrix as the `CliffSER2D` model. However, instead of expanding features, its four 2D-Clifford convolutional neural layers simply filter the input element $\boldsymbol{x} \in Cl^{H\times W}_{p,q}$ and reduce its spatial resolution, by setting the stride hyperparameter $S \geq 2$, leading to a dimensions $H_4 \times W_4$:

$$l_{1-4} : Cl^{H\times W}_{p,q} \rightarrow Cl^{H_4\times W_4}_{p,q}, \qquad \boldsymbol{x} \mapsto \boldsymbol{z}^{(l_4)} \quad . \tag{20}$$

No activation function and average pooling are employed. Each pixel $z^{(l_4)}(i, j)$, where $i = 1, \ldots, H_4$ and $j = 1, \ldots, W_4$, is rearranged into a single ordered array $z^{(l_4)}(c)$ with $c = 1, \ldots, H_4 \times W_4$. The subsequent three Clifford neural layers act as a learnable spatial downsampling, mapping the data as follows:

$$l_5, l_6, l_7 : Cl^{H_4\times W_4}_{p,q} \rightarrow Cl^{M}_{p,q} \rightarrow Cl^{N}_{p,q} \rightarrow Cl^{O}_{p,q} \tag{21}$$

$$\boldsymbol{z}^{(l_5)} \mapsto \boldsymbol{z}^{(l_6)} \mapsto \boldsymbol{z}^{(l_7)} \mapsto \boldsymbol{z}^{(l_8)} \quad , \tag{22}$$

The final mapping reduces O algebra elements to six elements, which are then converted into six scalars through a classical linear layer. The graphical representation of this model is depicted in fig. 9.

5) `CliffW2V`: The `CliffW2V` model processes raw audio input using the "wav2vec" backbone architecture as an encoder. This is followed by a projection head consisting of two Clifford linear layers, performing the mapping:

$$l_1, l_2 : Cl^{512}_{p,q} \rightarrow Cl^{M}_{p,q} \rightarrow Cl^{6}_{p,q} \tag{23}$$

$$\boldsymbol{z}^{(l_1)} \mapsto \boldsymbol{z}^{(l_2)} \mapsto \boldsymbol{z}^{(l_3)} \quad . \tag{24}$$

Eventually, a traditional linear layer converts $\boldsymbol{z}^{(l_3)} \in Cl^{6}_{p,q}$ into six scalar outputs.

IV. Experiments

A. Dataset

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [22] consists of 7356 recordings featuring acted emotional expressions by 24 professional actors (12 female, 12 male), delivering two lexically-matched statements

in a neutral North American accent. The dataset is organized into three modalities: full audio-visual (AV), video-only, and audio-only, as well as two vocal channels: speech and song. This study specifically focuses on the raw audio-only wav files. Each recording includes an actor expressing one of eight emotions: calm, neutral, happy, sad, angry, fearful, surprised, and disgusted, with two levels of emotional intensity (normal and strong).

B. Experimental setup and model selection

The experimental pipeline comprise the following steps: audio signals pre-processing and EMD analysis, feature extraction, feature clustering, model training, model evaluation.

1) Audio signals pre-processing and EMD analysis.: For supervised approaches, audio data were filtered using a fourth-order Butterworth low-pass filter with a cutoff frequency of 8 kHz. For end-to-end self-supervised models, audio files were resampled to 16 kHz and converted to mono, enabling the wav2vec backbone to produce 512-dimensional latent representations. To ensure consistent spatial dimensions, data for 2D models were zero-padded to match the length of the longest recording in the dataset. EMD analysis, conducted only for feature-based models, extracted the first 10 IMFs and grouped them into High Frequency (HF), Medium Frequency (MF), Low Frequency (LF), and Very Low Frequency (VLF) components.

2) Feature extraction: Both logMel spectrogram and MFCCs were extracted from the pre-processed audio signal and from each single EMD group of HF, MF, LF, VLF signals, separately. For 2D processing, features were discretized into 128 frequency bins and 188 short-time analysis points. For 1D processing, the temporal dimension was averaged into a single compressed representation with 128 frequency points. Feature extraction was performed using the librosa's "feature.mfcc" and "feature.melspectrogram" functions.

3) Feature clustering: Feature clustering was applied to both MFCC and logMel spectrogram features to assess their separability into six classes corresponding to the six emotional states. This step provides insights into the effectiveness of the features, demonstrating whether the latent patterns can be exploited for efficient classification. Initially, PCA reduced the data dimensionality to 50 components, preserving the majority of the variance. Further reduction to a 2D space was achieved using t-SNE, followed by K-means clustering, with $K = 6$.

4) Model selection and training: Model selection employed a Grid Search technique, and model stability was evaluated using 5-fold cross-validation. For the "pure" Clifford models (`PureCliffSER1D`, `PureCliffSER2D`), the structure comprised the following layers: $[1, 1, 1, 1, 256, 128, 64]$, indicating no feature expansion in the convolutional layers. Downsampling was achieved with $M = 256$, $N = 128$ and $O = 64$, following the graphical model depicted in fig. 9. For the (`CliffSER1D` and `CliffSER2D`) models, the structure comprised the following channels dimensionality: $[32, 64, 128, 256, 128, 64]$. `CliffSER2D` graphical model is reported in fig. 8.

For 2D models, the Grid Search tested convolution kernel sizes (3 or 5) and algebra signatures $[(2, 0), (0, 2), (3, 0), (0, 3)]$. Stride values of 1 or 2 were tested.

All models, except `Wav2Vec` and `CliffW2V`, were trained on MFCCs or logMel features, both with and without EMD decomposition. For the end-to-end `CliffW2V`, the Grid Search parameters included pooling methods ("mean" or "max"), hidden layer dimensions (100 or 200), dropout rates (0.3 or 0.5), and algebra signatures.

To address class imbalance in the 5-fold training sets, data augmentation was performed using the Audiomentations library, applying transformations such as Gaussian noise, time stretching, and pitch shifting to augment data and avoid leakage.

Traditional models (`1DCNN`, `2DCNN`, `Wav2Vec`) served as baselines for comparison with the Clifford algebra-based models.

5) Model evaluation: In the experiments, we evaluated the performance of our models using four key metrics: average accuracy, average precision, average recall, and average F1-score:

- **Average accuracy** is the ratio of correctly predicted instances to the total instances.
- **Average precision** assesses the proportion of true positive predictions among all positive predictions, indicating the model's ability to avoid false positives.
- **Average recall** evaluates the proportion of true positive predictions among all actual positive instances, reflecting the model's capability to identify all relevant cases.
- **F1 score** combines precision and recall into a single metric, providing a balanced measure of the model's accuracy, especially when dealing with imbalanced datasets.

V. Results

In this section we highlight the main results achieved by this work. Results concerning 1D data are summarized in TABLE I, while TABLE II reports results for 2D input. Self-supervised methods' results, for the AER downstream task, are reported in TABLE III.

Preliminary to any further discussion, clustering results, as evidenced by the confusion matrix, the overall accuracy of 18.37% and the plots in fig. 10, fig. 11 and fig. 12 reveal substantial challenges in distinguishing between the six emotion classes, regardless of whether MFCCs or logMel spectrograms are used as features. The high degree of misclassification across all clusters suggests that emotion classification on the RAVDESS dataset is not easily solvable by simple linear methods. The 2D latent space produced by concatenating PCA and t-SNE methods is not linearly separable, indicating that more complex models or more representative features are required to achieve good performances on this benchmark.

In light of this observation, it is notable how the utilization of Clifford algebra provides a significant, albeit modest, positive contribution to the audio emotion recognition task compared to traditional architectures in single-step supervised

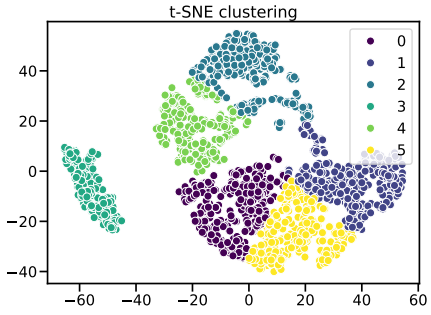

Fig. 10. Scatter plot clustering

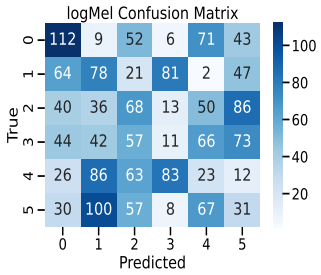

Fig. 11. logMel Confusion Matrix

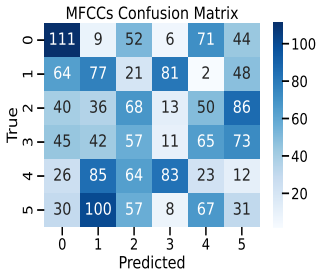

Fig. 12. MFCCs Confusion Matrix

TABLE I
RESULTS FOR SPEECH EMOTION RECOGNITION ALGORITHMS

		CliffSER1D						PureCliffSER1D					
		Algebra	Avg. Train Acc.	Avg. Test Acc.	Avg. Test Precision	Avg. Test Recall	Avg. Test F1-Score	Algebra	Avg. Train Acc.	Avg. Test Acc.	Avg. Test Precision	Avg. Test Recall	Avg. Test F1-Score
EMD	MFCCs	$Cl_{3,0}$	81.16%	48.53% ± 2.01	0.50	0.49	0.49	$Cl_{2,0}$	72.54%	59.27% ± 0.04	0.61	0.59	0.59
	logMEL	$Cl_{0,3}$	80.45%	51.32% ± 2.23	0.51	0.51	0.51	$Cl_{2,0}$	72.81%	59.66% ± 1.73	0.61	0.60	0.60
NO EMD	MFCCs	$Cl_{3,0}$	81.22%	55.39% ± 4.21	0.55	0.55	0.55	$Cl_{2,0}$	65.81%	55.42% ± 3.68	0.55	0.55	0.55
	logMEL	$Cl_{0,3}$	88.65%	**65.82%± 4.99**	**0.68**	**0.66**	**0.66**	$Cl_{0,2}$	64.66%	58.92% ± 0.03	0.60	0.59	0.59
1DCNN		MFCCs	90.34%	59.18% ± 4.27	0.63	0.59	0.59	logMEL	88.71%	59.93% ± 3.16	0.60	0.59	0.60

TABLE II
RESULTS FOR SPEECH EMOTION RECOGNITION ALGORITHMS

		CliffSER2D						PureCliffSER2D					
		Algebra	Avg. Train Acc.	Avg. Test Acc.	Avg. Test Precision	Avg. Test Recall	Avg. Test F1-Score	Algebra	Avg. Train Acc.	Avg. Test Acc.	Avg. Test Precision	Avg. Test Recall	Avg. Test F1-Score
EMD	MFCCs	$Cl_{2,0}$	94.95%	69.26% ± 1.86	0.71	0.69	0.69	$Cl_{0,2}$	77.29%	58.50% ± 2.18	0.59	0.59	0.58
	logMEL	$Cl_{0,3}$	95.87%	70.86% ± 4.21	0.71	0.71	0.71	$Cl_{0,2}$	79.39%	64.50% ± 2.06	0.65	0.65	0.64
NO EMD	MFCCs	$Cl_{0,3}$	96.00%	75.26% ± 2.11	0.76	0.75	0.76	$Cl_{2,0}$	75.24%	60.58% ± 3.05	0.61	0.61	0.60
	logMEL	$Cl_{0,3}$	95.99%	**78.78% ± 0.86**	**0.80**	**0.79**	**0.79**	$Cl_{0,3}$	74.90%	60.88 ± 0.04	0.62	0.61	0.61
2DCNN		MFCCs	91.18%	75.05% ± 2.36	0.76	0.75	0.75	logMEL	92.57%	76.35% ± 2.16	0.77	0.76	0.76

TABLE III
RESULTS FOR SPEECH EMOTION RECOGNITION ALGORITHMS

	Algebra	Avg. Train Acc.	Avg. Test Acc.	Avg. Test Precision	Avg. Test Recall	Avg. Test F1-Score
CliffW2V	$Cl_{0,3}$	99.37%	84.13% ± 0.07	0.84	0.84	0.84
Wav2Vec	-	98.64%	**90.65% ± 0.01**	**0.90**	**0.90**	**0.90**

learning scenarios. The performance improvement is evident for both 1D and 2D input cases fig. 13, fig. 14. Specifically, the CliffSER1D model, which receives log-Mel spectrograms as input, achieves an estimated expected accuracy of $65.82\% \pm \mathbf{4.99}$, compared to $59.93\% \pm 3.16$ for the traditional version of the same architecture 1DCNN. Moreover, CliffSER1D also outperforms 1DCNN in precision (0.68 vs. 0.60), recall (0.66 vs. 0.59) and F1-Score (0.66 vs. 0.60). Similarly, the CliffSER2D model, utilizing log-Mel spectrograms as input, achieves an average accuracy of $78.78\% \pm 0.86$ superior to the $76.35\% \pm 0.86$ achieved by 2DCNN, with precision (0.80 vs. 0.77), recall (0.79 vs. 0.76) and F1-Score (0.79 vs. 0.76) also showing better performance. Confusion matrix for the CliffSER2D model is depicted in fig. 16.

The CliffSER1D and CliffSER2D models outperform both PureCliffSER1D, PureCliffSER2D, and their corresponding traditional counterparts 1DCNN and 2DCNN. Conversely, PureCliffSER1D and PureCliffSER2D exhibit inferior performance compared to 1DCNN and 2DCNN.

The optimal feature for the task is undoubtedly the log-Mel spectrogram, in both the 1D and 2D domains. Clifford and classical architectures, receiving log-Mel data as input, achieve superior performance compared to those ingesting 2D MFCC spectrograms or time-averaged 1D spectrograms. This holds true even when the data has been pre-processed using EMD analysis, as evidenced by CliffSER2D ($70.86\% \pm 4.21$ vs $69.26\% \pm 1.86$).

Surprisingly, models processing EMD-preprocessed data struggle to compete with Clifford-based approaches that receive scalar values as input (NO EMD).

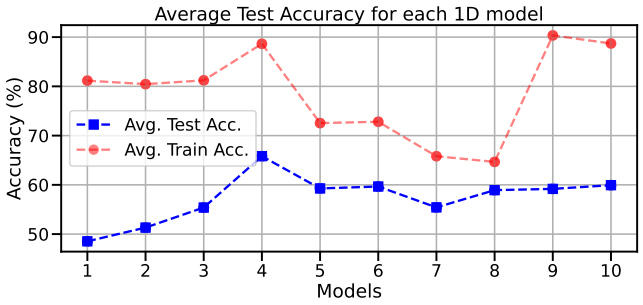

Fig. 13. Accuracy 1D plot.

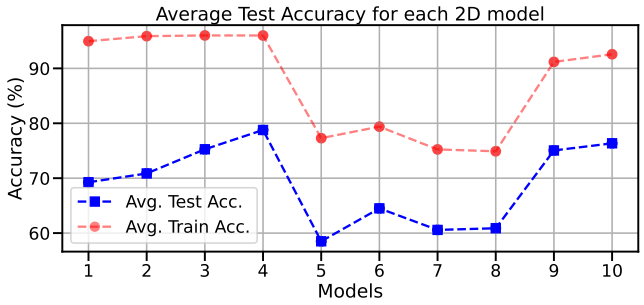

Fig. 14. Accuracy 2D plot.

		ALGEBRA		KERNEL		STRIDE	
N°	NAME	1D	2D	1D	2D	1D	2D
1	EMD MFCCs CliffSER	3,0	2,0	15	5	1	2
2	EMD logMEL CliffSER	0,3		15	5	1	2
3	NO EMD MFCCs CliffSER	3,0	0,3	15	5	1	2
4	NO EMD logMEL CliffSER	0,3		15	5	1	2
5	EMD MFCCs PureCliffSER	2,0	0,2	15	5	1	2
6	EMD logMEL PureCliffSER	2,0	0,2	15	5	1	2
7	NO EMD MFCCs PureCliffSER	2,0		15	5	1	2
8	NO EMD logMEL PureCliffSER	0,2	0,3	15	5	1	2
9	CNN MFCCs	–		15	5	1	2
10	CNN logMEL	–		15	5	1	2

Fig. 15. Legend and best hyperparameters' values.

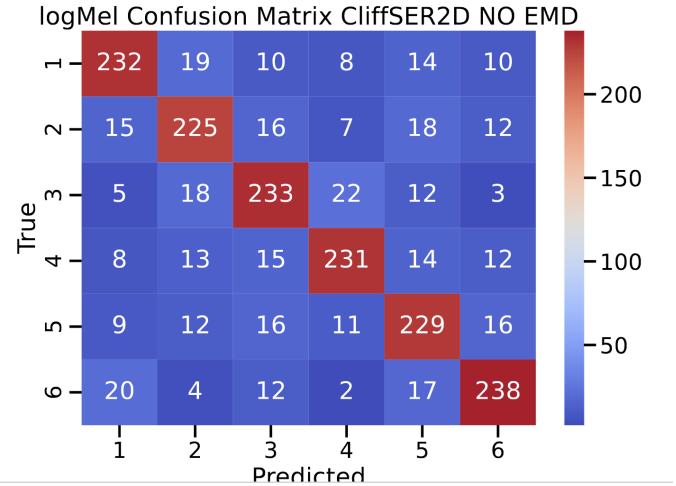

Fig. 16. `CliffSER2D` confusion Matrix (NO EMD, logMel).

For `CliffSER1D` and `CliffSER2D` models, the most effective algebraic setting is $Cl_{0,3}$. Conversely, for models closely aligned with geometric-algebraic features (`PureCliffSER1D` and `PureCliffSER2D`), the preferred signatures are primarily $Cl_{0,2}$ and $Cl_{2,0}$. The `CliffW2V` model favors the algebra $Cl_{2,0}$, but its performance falls short compared to the `Wav2Vec` model with a traditional classifier as the projection head. Specifically, the former achieves an accuracy of $84.14\% \pm 0.07$, while the latter achieves $90.65\% \pm 0.01$.

VI. Discussion and Conclusions

Our results demonstrate that EMD analysis does not produce multichannel data with beneficial algebraic-geometric information for the AER task. Representing data in a four-dimensional space using EMD does not improve performance, regardless of which feature representation (MFCCs or log-Mel spectrograms) is employed to represent each IMF group (HF, MF, LF, VFL). This may be due to EMD's inability to separate audio content distinguishing different expressions, such as happiness and disgust, into distinct IMF ranges. However, it is more probable that constraining the four IMF ranges as multi-vector parts fails to highlight useful intra- and inter-channel correlations for the Clifford networks. Despite IMFs representing different views of the same phenomenon, they lack a defined algebraic-geometric characterization. They are merely four sets of scalar values with spatial organization (MFCC coefficients or log-Mel spectrogram frequency points) and 1D temporal granularity. During training, the proposed Clifford models prefer constructing hypercomplex parts from scratch rather than managing the original signal's IMF decomposition. In general, it would be useful to explore representing audio data as pairs, triples, or quadruples of temporally and/or spatially organized values that naturally fit into an algebraic space as a hypercomplex number.

Overall, the `CliffSER1D` and `CliffSER2D` models outperform the `PureCliffSER1D` and `PureCliffSER2D` models. This indicates that applying feature expansion and external non-linear activation functions is necessary for better performance when the input lacks a natural algebraic representation.

When maximizing the geometric bias of the algebraic framework (i.e., using `PureCliffSER1D` or `PureCliffSER2D`), the best signatures identified by grid search are mostly four-dimensional (i.e., $Cl_{2,0}$ or $Cl_{0,2}$). This means that, taking into account only the "pure" models, Clifford neural networks with fewer parameters (i.e. less model complexity) perform better. However, those models are worse than the`CliffSER1D`, `CliffSER2D`,

and `CliffW2V` models, which typically select eight-dimensional algebras ($Cl_{3,0}$ or $Cl_{0,3}$), i.e signatures with double the parameters of $Cl_{2,0}$ and $Cl_{0,2}$. Although $Cl_{0,3}$ can incorporate rototranslations as learnable elements, models using feature expansion and external non-linearities might prefer this signature due to the higher number of parameters, offering greater complexity and flexibility for the task, while ignoring the simple geometrical interpretability of the "pure" way of using lower-dimensional algebras.

The bias introduced by the network is not ideal for this task, especially in the context of self-supervised learning. A pre-trained model that doesn't explicitly create a latent space aligned with the Clifford algebra formalism is unsuitable for using Clifford linear layers as the main components of the projection head, regardless of the downstream task.

However, Clifford algebra is a valuable tool for "creating" geometrically and algebraically relevant information layer by layer. When the coefficients of the multivector parts, aside from the basic scalar unit, are set to zero (and not filled with EMD values), the `CliffSER1D` and `CliffSER2D`, especially when fed by 1D or 2D log-Mel features, outperform other models. Essentially, Clifford algebra acts as a version with more weights compared to traditional layers. This can enhance performance for spatially and temporally structured data, but in self-supervised learning contexts, it may lead to overfitting.

REFERENCES

[1] Akçay, Mehmet Berkehan, and Kaya Oğuz. "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers." Speech Communication 116 (2020): 56-76.

[2] Keltner, Dacher, et al. "Emotional expression: Advances in basic emotion theory." Journal of nonverbal behavior 43 (2019): 133-160.

[3] Picard, Rosalind W. Affective computing. MIT press, 2000.

[4] P. Ekman et al., "Basic emotions," Handbook of cognition and emotion, vol. 98, no. 45-60, p. 16, 1999.

[5] J. A. Russell, "A circumplex model of affect." Journal of personality and social psychology, vol. 39, no. 6, p. 1161, 1980.

[6] Daily, Shaundra B., et al. "Affective computing: historical foundations, current applications, and future trends." Emotions and affect in human factors and human-computer interaction (2017): 213-231.

[7] Panda, Renato, Ricardo Malheiro, and Rui Pedro Paiva. "Audio features for music emotion recognition: a survey." IEEE Transactions on Affective Computing 14.1 (2020): 68-88.Science, 1989.

[8] Lopez, E., Grassucci, E., Capriotti, D., & Comminiello, D. (2024). Towards Explaining Hypercomplex Neural Networks. arXiv preprint arXiv:2403.17929.

[9] Brandstetter, J., Berg, R. V. D., Welling, M., & Gupta, J. K. (2022). Clifford neural layers for pde modeling. arXiv preprint arXiv:2209.04934.

[10] Torralba, A., Isola, P., & Freeman, W. T. (2024). Foundations of Computer Vision. MIT Press.

[11] Comminiello, D., Grassucci, E., Mandic, D. P., & Uncini, A. (2024). Demystifying the Hypercomplex: Inductive Biases in Hypercomplex Deep Learning. arXiv preprint arXiv:2405.07024.

[12] Rabiner, L., & Schafer, R. (2010). Theory and applications of digital speech processing. Prentice Hall Press.

[13] Lounesto, P.: Clifford algebras and spinors. In: Clifford Algebras and Their Applications in Mathematical Physics, pp. 25–37. Springer (2001)

[14] Dorst, L., Fontijne, D., Mann, S.: Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2009)

[15] Huang, Norden E., et al. "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis." Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences 454.1971 (1998): 903-995.

[16] Krishnan, Palani Thanaraj, Alex Noel Joseph Raj, and Vijayarajan Rajangam. "Emotion classification from speech signal based on empirical mode decomposition and non-linear features: Speech emotion recognition." Complex & Intelligent Systems 7 (2021): 1919-1934.

[17] Quinn, A. J., Lopes-dos-Santos, V., Dupret, D., Nobre, A. C., & Woolrich, M. W. (2021). EMD: Empirical mode decomposition and Hilbert-Huang spectral analyses in Python. Journal of open source software, 6(59).

[18] Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., & Tang, J. (2021). Self-supervised learning: Generative or contrastive. IEEE transactions on knowledge and data engineering, 35(1), 857-876.

[19] Wang, T., & Isola, P. (2020, November). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In International conference on machine learning (pp. 9929-9939). PMLR.

[20] Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. arXiv preprint arXiv:1904.05862.

[21] Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in neural information processing systems, 33, 12449-12460.

[22] Livingstone, S. R., & Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [Data set]. In PLoS ONE (1.0.0, Vol. 13, Numero 5, pag. e0196391). Zenodo. https://doi.org/10.5281/zenodo.1188976