Learning Speech Emotion Representations in the Quaternion Domain

Eric Guizzo, Tillman Weyde, *Member, IEEE*, Simone Scardapane, and Danilo Comminiello, *Senior Member, IEEE*

***Abstract*—The modeling of human emotion expression in speech signals is an important, yet challenging task. The high resource demand of speech emotion recognition models, combined with the general scarcity of emotion-labelled data are obstacles to the development and application of effective solutions in this field. In this paper, we present an approach to jointly circumvent these difficulties. Our method, named RH-emo, is a novel semi-supervised architecture aimed at extracting quaternion embeddings from real-valued monoaural spectrograms, enabling the use of quaternion-valued networks for speech emotion recognition tasks. RH-emo is a hybrid real/quaternion autoencoder network that consists of a real-valued encoder in parallel to a real-valued emotion classifier and a quaternion-valued decoder. On the one hand, the classifier permits to optimization of each latent axis of the embeddings for the classification of a specific emotion-related characteristic: valence, arousal, dominance, and overall emotion. On the other hand, quaternion reconstruction enables the latent dimension to develop intra-channel correlations that are required for an effective representation as a quaternion entity. We test our approach on speech emotion recognition tasks using four popular datasets: IEMOCAP, RAVDESS, EmoDB, and TESS, comparing the performance of three well-established real-valued CNN architectures (AlexNet, ResNet-50, VGG) and their quaternion-valued equivalent fed with the embeddings created with RH-emo. We obtain a consistent improvement in the test accuracy for all datasets, while drastically reducing the resources' demand of models. Moreover, we performed additional experiments and ablation studies that confirm the effectiveness of our approach.**

***Index Terms*—Speech emotion recognition, quaternion neural networks, quaternion algebra, transferable embeddings.**

I. Introduction

HUMAN-MACHINE interaction is becoming increasingly important in our everyday life. Research on speech recognition reached near-human performance in recent years. Nevertheless, besides the mere sequence of words, there is additional information that the speech can carry, in particular about emotion. Speech Emotion Recognition (SER) is therefore acquiring a growing role in research on human-machine interaction, since it helps provide a more complete account of the information conveyed by speech signals. Despite the impressive success that neural networks have achieved in this task, SER is still challenging due to the variability of emotional expression, especially in real-world scenarios where generalization to unseen speakers and contexts is required [1], [2]. The difficulty of this task is partly related to the general scarcity of emotionally-labelled audio data, which is due to the high cost of recording and labelling such data. Another well-known difficulty of SER is that emotional information in speech involves long-term temporal dependencies that are in the order of seconds [3], [4], [5]. This forces models to analyze large temporal windows and, consequently, to use a large number of resources.

This study proposes a joint solution for two main issues in SER research. Broadly speaking, we propose to map speech signals into a compact multi-channel latent representation that permits having different "emotional viewpoints" of the signal, which are signal representations individually related to different components of human emotion, namely: valence arousal and dominance. To this end, we make use of quaternion information processing, which is a well-established strategy to minimize models' resource demand without reducing their performance, as we discuss in detail in Sections II and III. The resulting proposed model, named Real to Emotional H-Space (RH-emo), is a semi-supervised autoencoder architecture that maps input speech signals to an embedded emotional-quaternion space. The axes of the embedded space are individually related to different emotion characteristics, i.e., valence, arousal, and dominance, which are represented as quaternion components. As we will be further explored from Section V onward, when used as a feature extractor that feeds into quaternion neural networks (QNNs), RH-emo improves the performance in SER tasks while considerably reducing the number of trainable parameters and computing resources, compared to equivalent real-valued models processing plain spectrograms. This behavior is also consistent in situations where data is very scarce.

The specific contributions of our work are the following:

- We define a novel method, RH-emo, that draws quaternion-valued embeddings from speech signals, where each

Manuscript received 5 April 2022; revised 2 December 2022 and 27 January 2023; accepted 9 February 2023. Date of publication 1 March 2023; date of current version 16 March 2023. This work has been performed while the first author was a Ph.D. Visiting Student at Sapienza University of Rome, Italy. This work was supported by the "Progetti di Ricerca Grandi" of Sapienza University of Rome under Grant RG11916B88E1942F. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sakriani Sakti. *(Corresponding author: Eric Guizzo.)*

Eric Guizzo and Tillman Weyde are with the Department of Computer Science, City, University of London, Northampton Square, London EC1V 0HB, U.K. (e-mail: Eric.Guizzo@city.ac.uk; t.e.weyde@city.ac.uk).

Simone Scardapane and Danilo Comminiello are with the Department of Information Engineering, Electronics and Telecommunications (DIET), Sapienza University of Rome, 00184 Rome, Italy (e-mail: simone.scardapane@uniroma1.it; danilo.comminiello@uniroma1.it).

The RH-emo repository is available at: https://github.com/ispamm/rhemo.

Digital Object Identifier 10.1109/TASLP.2023.3250840

quaternion component is tailored to a specific emotional characteristic.
- We leverage the capabilities of quaternion emotion embeddings and the effectiveness of quaternion convolutional neural networks (QCNNs) to jointly solve two of the most significant issues related to speech emotion recognition: data scarcity and high resource demand.
- We extensively evaluate our approach using 4 popular SER datasets and 3 widely-used CNN-based architectures.
- We provide open-source code[1] and pretrained models[2] that can be exploited to improve the performance and efficiency of existing SER models.

The remainder of the paper is organized as follows: Section II reviews the relevant literature, Section III is a brief overview of quaternion neural networks, Section IV describes our proposed method in detail, Section V presents our experimental setup and results, Section VI presents the ablation studies we conduct, Section VII discusses our outcomes and the properties of our approach and Section VIII draws the conclusions of this paper.

II. Background

In the literature, two main approaches to labeling expressed human emotions exist. On the one hand, discrete models provide a set of fixed emotion categories, such as *happy, sad, angry, fearful, surprised, disgusted, neutral*. On the other hand, continuous models map emotions into a multidimensional space. The most common model is a 2D *valence-arousal* space, where valence describes the degree of emotional pleasantness and arousal (or activation) of the intensity of the emotion. Dominance can be added as a third dimension describing the amount of control of a person expressing an emotion. This encodes a so-called valence-arousal-dominance space [6], [7], [8]. Discrete emotions can be mapped in this continuous space although the exact mapping is not standardized and different studies can use slightly different mappings.

A traditional approach to SER is based on two consecutive stages: hard-coded extraction of affect-salient features followed by a learning-based classification or regression. Various combinations of features and classifier types have been proposed. The most commonly used features are: base pitch, formant features, energy/spectral features, and prosody. A wide variety of classifiers has been proposed: artificial neural networks [9], [10], [11], Bayesian networks, [12], Hidden Markov Models [13], [14], support vector machines [15], [16], and Gaussian mixture models [17]. Nevertheless, in state-of-the-art methods, there is no default choice of features and classifier type [18]. With the advent of deep learning, end-to-end learning mostly replaced hard-coded feature extraction and selection, with models automatically extracting features from low-level representations of the input data (usually Fourier-based transforms, wavelet transforms, or raw audio data). This enables a model to fine-tune the feature extraction for a specific task and, consequently, often obtain a higher accuracy compared to engineered feature extraction. A range of deep learning architectures have been adopted for SER. The most commonly used are convolutional neural networks [19], [20], [21], recurrent neural networks [22], [23] and combinations of the two [24], [25], [26] Various studies directly compare the performance of approaches using end-to-end learning and hard-coded feature extraction, showing that the former generally provides a higher classification accuracy on the same data [27], [28], [29], [30]. Nevertheless, as a drawback, deep learning models generally require a higher computational cost and longer training times than traditional machine learning techniques and the end-to-end learning usually requires a large number of labelled data [31], [32].

A well-established solution to overcome the data scarcity in SER is transfer learning by weight initialization: network weights are initialized with values from a network that was pretrained with a different task, possibly on a different (usually large) dataset. Many variants of this method have been shown to improve the performance of SER models in limited-data scenarios and even when the task is rather distant from speech emotion [33], [34], [35]. Also, various data augmentation strategies have been successfully adopted for the same purpose, e.g. [36], [37]. On the other hand, the application of dimensionality reduction transformations to the model's input data is an established strategy for reducing resource demands while limiting the loss of useful information carried by the input data. Among others, autoencoders, PCA-based approaches, and transformer networks have been used in the field of SER [34], [38], [39], obtaining improvement both in the model's efficiency and classification accuracy.

A recent and increasingly popular strategy to improve the efficiency and the performance of deep learning models is the use of quaternion information processing [40], [41], [42], [43], [44], [45], [46]. Performing operations in the quaternion domain permits bootstrap intra-channel correlations in multi-dimensional signals [47], [48], i.e., among the color channels of RGB-encoded images. Moreover, due to the fewer degrees of freedom of the Hamilton product compared to the regular dot product, quaternion networks have a significantly lower number of parameters compared to the real counterparts [40]. Quaternion-valued neural networks have also been successfully adopted in the audio domain [49], [50] and specifically for speech recognition [45] and speech emotion recognition [46]. Nevertheless, an intrinsic limitation of quaternion information processing is that it requires three or four-dimensional data as input, where intra-channel correlations exist [41], [42], [43], [44]. This is necessary to enable the benefits derived from the use of the Hamilton product instead of the regular dot product, as further discussed in Section III. In the audio domain, first-order Ambisonics [51] signals are naturally suited for a quaternion representation, being four-dimensional and presenting strong correlations among the spatial channels, and the application of quaternion networks to problems related to this audio format has already been extensively investigated [50], [52], [53], [54].

[1][Online]. Available: https://github.com/ispamm/rhemo.

[2][Online]. Available: https://drive.google.com/drive/folders/1BWvbxqnsHK7FyXB1_L_DlO6UFECkNRvz?usp=sharingPretrained models: rhemo/weights.

Nevertheless, in the vast majority of cases, audio-related machine-learning tasks deal with monaural signals, which are usually treated as vectors of scalars (time-domain signals), matrices of scalars (magnitude spectrograms), or 3D tensors (complex spectrograms). Hence they can not be naturally represented as a quaternion entity and additional processing is required to produce a suitable quaternion representation of these signals.

A number of different approaches have been proposed to overcome the necessity of having three or four-dimensional input data with intra-channel correlations. Among others, [45] use Mel spectrograms, cepstral coefficients, and first and second-order derivatives as the four axes of the encoded quaternion. In contrast, [46] convert Mel spectrograms to color-scaled images and use the RGB channels as axes of the encoded quaternion, following a computer vision-oriented approach. Parcollet et al. [55] presented two learning-based approaches to map real-valued vectors into the quaternion domain, by producing through a network four-channel representations of the input data that present meaningful intra-channel correlations. On the one hand, the Real to H-space encoder [55], applied to speech recognition tasks, consists of a simple real-valued dense layer applied at the beginning of a quaternion classifier network, which is trained jointly with the classifier. On the other hand, the Real to H-space Autoencoder, tested in the natural language processing field (conversation theme identification) [55] operates in an unsupervised way. Such a method contains a real-valued encoder and a quaternion-valued decoder, where the latter is expected to enable both the network's embeddings and output to present meaningful intra-channel correlations that can be exploited by a quaternion-valued classifier network.

In this paper, we introduce RH-emo, a hybrid real-quaternion autoencoder-classifier architecture that is trained in a semi-supervised fashion in order to optimize each axis of the embedding dimension to different emotional characteristics: the first channel is optimized for discrete emotion recognition and the 3 other channels are individually optimized for the classification of valence, arousal, and dominance (as shown in Fig. 1). RH-emo is intended to be used as a feature extractor that permits using QNNs for SER tasks with real-valued signals without additional preprocessing. This approach has two advantages: it improves the performance of SER models even in situations where data is scarce and it drastically reduces the number of network parameters, consequently reducing the resource demand. We extend the approach of the quaternion autoencoder in [55] by specializing the learned quaternion representation for our specific task (SER), where the different axes are optimized for the detection of different emotional characteristics that are coherent with the most used criteria of emotion classification. Moreover, we implement it with a more complex architecture (deep convolutional autoencoder) and we apply it to a different domain: emotion recognition from speech audio.

III. Quaternion Convolutional Neural Networks

Operations between quaternion numbers are defined in the quaternions algebra $\mathbb{H}$. A quaternion Q is a four-dimensional extension of a complex number, defined as $\mathbf{q} = q_0 + q_1\hat{\imath} + q_2\hat{\jmath} + q_3\hat{\kappa} = q_0 + q$, where, q_0, q_1, q_2 are real numbers, and $\hat{\imath}$, $\hat{\jmath}$ and $\hat{\kappa}$ are the quaternion unit basis. In this representation q_0 is the real part and $q_1\hat{\imath} + q_2\hat{\jmath} + q_3\hat{\kappa}$ is the imaginary part, where $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{\kappa}^2 = -1$ and $\hat{\imath}\hat{\jmath} = -\hat{\jmath}\hat{\imath}$. From the latter assumption follows that the quaternion vector multiplication is not commutative. A quaternion can also be represented as a matrix of real numbers:

$$\mathbf{q} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}. \tag{1}$$

Analogously to real and complex numbers, a set of operations can be defined in the quaternion space:

- **Addition**: $\mathbf{q} + \mathbf{p} = (q_0 + p_0) + (q_1 + p_1)\hat{\imath} + (q_2 + p_2)\hat{\jmath} + (q_3 + p_3)\hat{\kappa}$
- **Conjugation**: $\mathbf{q}^* = q_0 - q_1\hat{\imath} - q_2\hat{\jmath} - q_3\hat{\kappa}$
- **Scalar multiplication**: $\lambda\mathbf{q} = \lambda q_0 + \lambda q_1\hat{\imath} + \lambda q_2\hat{\jmath} + \lambda q_3\hat{\kappa}$
- **Element multiplication** (or **Hamilton product**):

$$\begin{aligned} \mathbf{q} \otimes \mathbf{p} = {} & (q_0 + q_1\hat{\imath} + q_2\hat{\jmath} + q_3\hat{\kappa})\,(p_0 + p_1\hat{\imath} + p_2\hat{\jmath} + p_3\hat{\kappa}) \\ = {} & (q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3) \\ & + (q_0p_1 + q_1p_0 + q_2p_3 - q_3p_2)\,\hat{\imath} \\ & + (q_0p_2 - q_1p_3 + q_2p_0 + q_3p_1)\,\hat{\jmath} \\ & + (q_0p_3 + q_1p_2 - q_2p_1 + q_3p_0)\,\hat{\kappa}. \end{aligned} \tag{2}$$

The quaternion convolutional neural network (QCNN) is an extension of the real-valued convolutional neural network to the quaternion domain. For each input vector of a quaternion layer, the dimensions are split into four parts to compose a quaternion representation. In a quaternion-valued fully-connected layerthe parameters matrices are treated as a single quaternion entity with four components, even though they are manipulated as matrices of real numbers [56]. In a quaternion layer, the dot product operations used in real layers are replaced with the Hamilton product (2) between the input vector and a quaternion-represented weight matrix. This allows the processing of all input channels together as a single entity maintaining original intra-channels dependencies because the weights submatrices are shared among the input channels. Consequently, quaternion layers permit to spare the 75% of free parameters compared to their real-valued equivalents because, as shown in (2), the same components are re-used to build the output matrix.

In a QCNN, the convolution of a quaternion filter matrix with a quaternion vector is performed as the Hamilton product between the real-valued matrices representation of the input vector and filters. A quaternion convolution between a quaternion input vector$\mathbf{x} = x_0 + x_1\hat{\imath} + x_2\hat{\jmath} + x_3\hat{\kappa}$ and a quaternion filter $W = W_0 + W_1\hat{\imath} + W_2\hat{\jmath} + W_3\hat{\kappa}$ can be defined as:

$$W * x = \begin{bmatrix} W_0 & -W_1 & -W_2 & -W_3 \\ W_1 & W_0 & -W_3 & W_2 \\ W_2 & W_3 & W_0 & -W_1 \\ W_3 & -W_2 & W_1 & W_0 \end{bmatrix} * \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{3}$$

The optimization of quaternion-valued networks is identical to the one of a real network and can be achieved through regular

backpropagation. This is possible because of the use of split activation and loss functions, as introduced in [55], [57]. These functions map a quaternion-like entity back to the real domain, consequently enabling the use of standard loss functions for the network training.

IV. The Proposed RH-emo Model

A. Approach

The main aim of RH-emo is to map real-valued spectrograms to the quaternion domain, building compact emotion-related quaternion embeddings where each axis is optimized for a different emotional characteristic. In the embedded dimension, the real axis of the quaternion is optimized for the discrete classification of 4 emotions: *neutrality, anger, happiness, sadness* and the 3 complex axes are optimized for the prediction of emotion in a *valence, arousal* and *dominance* 3D space. This representation exploits the natural predisposition of quaternion algebra to process data where a 4 or 3-channels representation is meaningful. Nevertheless, in most machine learning applications of quaternion algebra, the input data is naturally organized with a meaningful shape, as happens for instance with RGB/RGBA images (where the color/alpha channels are treated as different quaternion axes) and first-order Ambisonics audio signals (where the 4 spatial channels are considered as the quaternion axes). In our case, instead, such quaternion representation is created through a semi-supervised learning procedure, where the different axes are forced to contain information related to different complementary emotion characteristics. Therefore, in a certain sense, the axes of this embedded dimension can be thought of as different "emotional points of view" of an audio signal.

RH-emo is intended to be used as a pretrained feature extractor to enable the use of quaternion-valued neural networks for SER tasks applied to monoaural audio signals. On the one hand, the emotion-related disentanglement among channels helps to enhance the performance of SER models, especially under conditions of data scarcity. Whereas, on the other hand, the reduced dimensionality together with the enabled possibility to classify the data with quaternion-valued networks permits to spare of a large number of network parameters, consequently lowering the resource demand and speeding up the training.

B. RH-emo Architecture

RH-emo is a hybrid real/quaternion autoencoder network. Its structure is similar to R2Hae [55], nevertheless, RH-emo is based on a convolutional design and it embraces multiple classification branches, as opposed to R2Hae. We used a public PyTorch implementation of convolution layers and operators.[3] As Fig. 1 shows, our RH-emo is composed of three components: an encoder $E(X)$ acting on the (real-valued) input spectrogram, producing an embedded vector. The output of the encoder is then fed separately to a (quaternion-valued) decoder $D(Z)$ to reconstruct the original spectrogram and to a classification

[3][Online]. Available: https://github.com/Orkis-Research/Pytorch-Quaternion-Neural-Networks

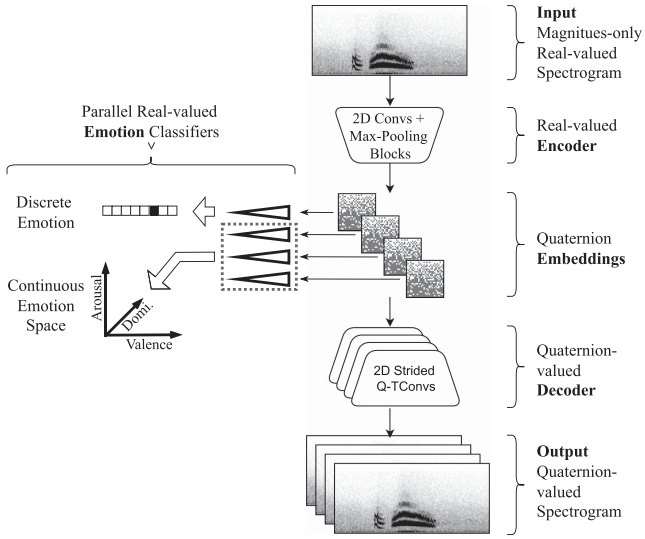

Fig. 1. RH-emo Block Diagram. An input magnitudes-only spectrogram is first propagated into a real-valued convolutional encoder that generates embeddings with a [4x64x64] shape. The network is then split into two branches: a completely unsupervised quaternion-valued decoder that reconstructs the input spectrogram projecting it in a four-channel quaternion space and a set of 4 parallel real-valued supervised classifiers, each connected to one of the four channels of the embeddings and separately classifying different emotion characteristics: discrete emotion, valence, arousal, and dominance.

head $C(Z)$ for performing emotion recognition. The classifier outputs four separate predictions y_D, y_v, y_a, and y_d which are, respectively, a discrete and a continuous (in the valence, arousal, dominance space) categorization of the emotional content of the spectrogram. The specific architecture for each of these blocks, as well as the loss function we optimize and the training strategy we adopt, is described more in detail in the following paragraphs.

1) Encoder: The input data, a magnitudes-only real-valued spectrogram in our case, is forward propagated through a real-valued autoencoder made up of 3 convolution blocks. Each block contains a 2D convolution layer (ReLU activations, 3x3 kernels, single-pixel stride, increasing channels number: 1, 2, 4), followed by max-pooling layers of dimension [2x2], [2x1], [2x1]. Moreover, only between the first and the second block, a batch normalization layer is present. The encoder produces an embedded vector that presents a dimensionality reduced by a factor of 0.25 compared to the input. In our experiments, we use input spectrograms with a shape of 1x512x128 (channels, time-steps, frequency-bins) and the embedded dimension created by the encoder has a shape of 4x64x64. The embedded vector is then forward propagated in parallel into four distinct real-valued classifiers and also into a quaternion-valued decoder. It is therefore important that the embedded vector contains a number of elements that is multiple of four, in order to be properly treated as a quaternion by the decoder section of the network.

2) Classifiers: Each classifier consists of a sequence of 3 real-valued fully connected layers, where the first 2 contain 4096 neurons and are followed by a dropout layer. In the first classifier, the output layer contains 4 output neurons (the number of emotional classes to be classified) and softmax activation.

Instead, the other 3 classifiers are identical and have one single output neuron with sigmoid activation, as they are individually aimed at a binary classification task: the prediction of "high" or "low" valence, arousal, and dominance, respectively.

3) Decoder: The decoder mirrors the encoder's structure but uses quaternion-valued 2D transposed convolutions with a stride that mirrors the pooling dimensions of the encoder, instead of the sequence of 2D real-valued convolutions and 2x2 max-pooling and a quaternion-valued batch normalization layer instead of its real-valued counterpart. The output of the decoder is therefore a matrix with the same dimensions as the input, but with 4 channels instead of a single one.

C. Loss Function

The loss function we minimize during the training of RH-emo is a weighted sum of the binary crossentropy reconstruction loss between the input spectrogram and the decoder's output, the categorical crossentropy classification loss of the emotion labels predicted by the supervised classifier in the middle of the network (discrete, valence and arousal).

The objective function we minimize is, therefore:

$$\mathcal{L} = \mathrm{BCE}_(X, Y_r) + \beta \cdot \{\mathrm{CE}_(p, t) + \alpha \cdot [\mathrm{BCE}_(v_p, v_t) + \mathrm{BCE}_(a_p, a_t) + \mathrm{BCE}_(d_p, d_t)]\} \tag{4}$$

where BCE is the binary crossentropy loss, CE is the categorical crossentropy loss, β and α are scalar weight factors, X is the input spectrogram, Y_r is the decoder's output re-mapped to the real domain through the split activation function (as discussed below), p and t are respectively the discrete emotion prediction and truth label, v_p/v_t, a_p/a_t and d_p/d_t are respectively the valence, arousal and dominance prediction, and truth labels.

For the reconstruction loss computation, it is necessary to map the quaternion-valued decoder output back to the real domain, in order to have the same shape as the input vector. For this purpose we use a stratagem similar to the "split activation" described in [55], [57]: we perform an element-wise mean across the channel dimension of the quaternion output, bringing back the 4-channels vector to a single-channel shape. During the training, this forces the model to not weigh the intra-channel correlations among the quaternion axes in the reconstruction term of the loss (the leftmost term of (4)). Our expectation is that this leaves room for the emotion recognition term of the loss (the rightmost term of (4)) for tuning these correlations, making them related to the emotional information.

D. Training Strategy

For the RH-emo training, we use the Interactive Emotional Dyadic Motion Capture Database (IEMOCAP) dataset [58], which includes: 5 speakers, 7529 utterances, 9:32 hours of audio, 10 emotion labels and it is in the English language. We selected this specific dataset for the following reasons: it is one of the most popular SER datasets, it contains a large number of datapoints, it is not limited to a restricted set of sentences, emotions are expressed by actors with a natural feeling rather than being over-emphasized [58] and it is labelled both in the discrete and continuous (valence, arousal, dominance) emotional domains.

We apply 4 preprocessing stages to the raw data: we first extract 4-second non-overlapped fragments (or zero-pad if a datapoint is shorter that this duration). Then, we compute the short-time Fourier transform (STFT) using 16 ms sliding windows with 50% overlap, applying a Hamming window and discarding the phase information. After this point, we normalize the whole dataset between 0 and 1 and, in the end, we zero-pad the spectrograms to match a shape of 512 (time-steps) x 128 (frequency-bins).

To permit proper convergence, we perform the training in 2 consecutive stages: we first train the network until convergence with the β weight set to 0. This removes the rightmost term from (4), consequently eliminating the emotion classification part of the loss. Doing so, we train the network in a completely unsupervised way only to perform a quaternion projection of the real input spectrogram, without taking into account any emotion-related information. After this stage, we re-train the network adding also the classification term in the loss in order to specialize the learned representations to the emotion recognition task, but also maintaining the embedded vector in a quaternion-compatible shape that is meaningful for the decoder part of the network. For this stage, we performed a grid search to find the best combination of the emotion classification weights β and α and we ended up using $\beta = 0.01$ and $\alpha = 100$. This means that overall we weigh more the reconstruction error in the loss function (thanks to the low β), and we weigh more the dimensional emotion classification compared to the discrete classification (thanks to the high α).

While for the first, completely unsupervised, training stage we use all data available with IEMOCAP, in the second supervised stage we use only a subset of the dataset, including only the datapoints related to 4 emotions (*angry, happy, neutral, sad)* and we merge the classes *happy* and *excited* as one single emotion class *happy*. This is a standard procedure with IEMOCAP, as the other labels contained in the dataset are highly imbalanced. For both training stages, we use subsets of approximately 70% of the data for training, 20% for validation, and 10% for the test set. We use a learning rate of 0.001 in the first stage and of 0.000001 in the second one, a batch size of 20 and the Adam optimizer [59]. We use dropout at 50% in the classification branches for the second training stage. We apply early stopping by testing at the validation loss improvement with patience of 100 epochs in the first stage and 30 epochs for the second one.

After these 2 training stages, we obtain a test reconstruction loss (the isolated leftmost term of (4)) of 0.00413 and competitive test classification accuracy: 60.7% for the discrete classification and respectively 65.4%, 75.3% and 70.2% for the valence, arousal, and dominance dimensions.

V. Evaluation

In order to test the capabilities and properties of RH-emo, we compare the classification accuracy for SER tasks obtained with real-valued CNN networks and equivalent quaternion-valued versions of them (QCNNs). For the quaternion versions we keep

the same architecture of the real CNNs, but we use quaternion-valued convolution and quaternion-valued fully connected layers instead of the canonical real-valued ones, with the exception of the final layer of the networks, which are real-valued also in the QCNNs. For the real networks, we use the magnitudes-only spectra as input, while for the quaternion networks we use the embeddings generated with RH-emo pretrained on IEMOCAP. Moreover, we compare and combine our approach with a standard transfer learning method performed on the same dataset (IEMOCAP): pretraining with weight initialization. Therefore we have two distinct types of pretraining: the pretraining of the RH-emo network, which we use to compute the emotional embeddings, and the pretraining of the CNNs that we use to perform the actual SER task. Both pretrainings are performed on the IEMOCAP dataset. To avoid confusion, from here on we will refer to the first as RH-emo pretraining and to the latter as CNN's pretraining.

Fig. 2 depicts all cases we include in our experimental setup. The color coding of Fig. 2 shows the 3 consecutive stages of our experiments: first, we pretrain RH-emo (yellow), then we pretrain the CNNs (orange) on IEMOCAP and finally we train or retrain the CNNs on other datasets. We have two types of baseline: the first one, shown in the upper row of Fig. 2, is a standard real-valued CNN with randomly-initialized weights. As a further baseline, as depicted in the second row of Fig. 2, we test a standard transfer learning approach applied to the real-valued CNNs: we pretrain on IEMOCAP (the same dataset used to train RH-emo) and we then initialize all weights of the SER CNNs but the ones of the final classification layer. The last two rows of Fig. 2, instead, show our approach, where we use RH-emo as a feature extractor to feed quaternion-valued CNNs. In the third row, only RH-emo pretraining happens, while in the last row both RH-emo and CNNs pretraining are performed. In the latter case, we first pretrain RH-emo, then we pretrain the CNN on IEMOCAP, and finally, we re-train the same CNN on different datasets.

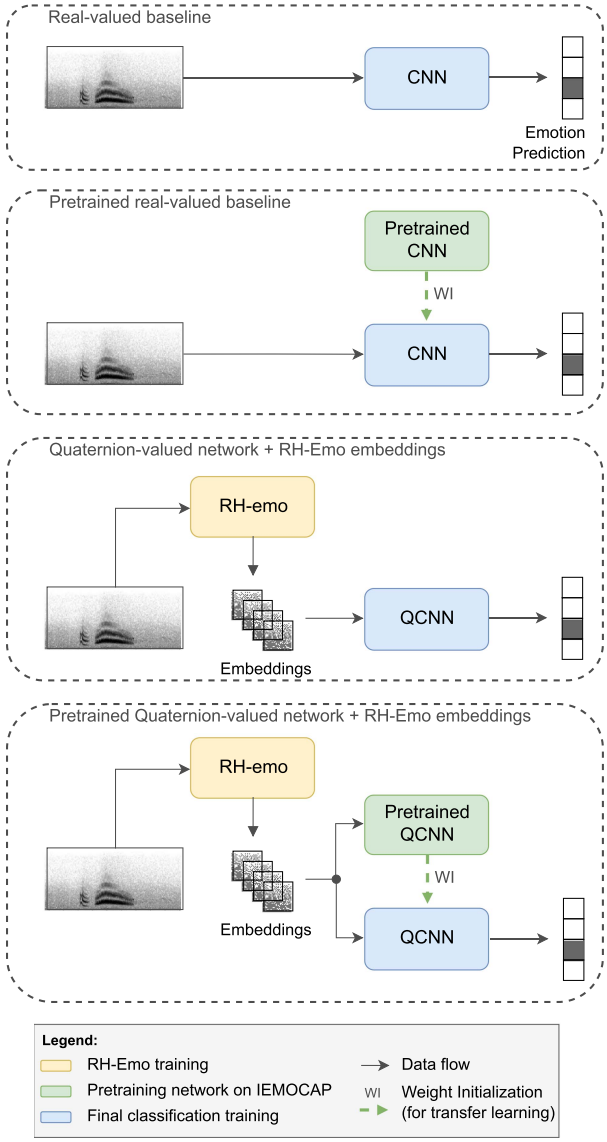

Fig. 2. Block diagram of our experimental setup. The yellow-to-blue color coding reflects 3 consecutive training stages. There are 2 separate pretraining stages: RH-emo pretraining (yellow) and CNN pretraining (green). The straight arrows indicate the data flow, while the dotted arrows, accompanied by the word WI, show where the weights of a pretrained network are used to initialize the initial weights of an identical network (transfer learning). The real-valued baseline is a regular CNN with random weight initialization, upper row. The pretrained real-valued baseline is the same network, but its weights are initialized with the ones of an identical network pretrained on IEMOCAP (the same dataset used to train RH-emo), second row. The quaternion-valued network is a quaternion-valued version of the real-valued baselines, in which (4 channel) input is generated by forward propagating the input spectrogram in RH-emo's encoder, third row. The pretrained quaternion-valued network is identical to the latter, but the weights of the CNN are initialized with the ones of an identical network pretrained on IEMOCAP, last row.

A. *Experimental Setup*

We evaluate RH-emo with 3 benchmark SER datasets:

1) RAVDESS, the Ryerson Audio Visual Database of Emotional Speech and Song [60]. 24 speakers, English language, 2542 utterances, 2:47 hours of audio, 8 emotion labels.
2) EmoDB, a Database of German Emotional Speech [61]. 10 speakers, German language, 535 utterances, 25 min of audio, 7 emotion labels.
3) TESS, the Toronto Emotional Speech Set [62]. 2 speakers, English language, 2800 utterances, 1:36 hours of audio, 7 emotion labels.

The preprocessing pipeline for these datasets is identical to the one we applied to IEMOCAP, as described in Section IV, except for the final normalization step. For the quaternion-valued networks we normalize data between 0 and 1 (as required by RH-emo), and for the real-valued networks we normalize to 0 mean and unity standard deviation to permit proper convergence.

We apply this approach to 3 popular CNN architectures with increasing capacity: VGG16 [63], AlexNet [64] and ResNet-50 [65], based on the Torchvision implementations[4]. These implementations present an adaptive average pooling

[4][Online]. Available: https://pytorch.org/vision/stable/_modules/torchvision.html

TABLE I
PRETRAINING RESULTS ON IEMOCAP

Arch.	Method	Params	Train acc.	Test acc.
RH-emo	/	1.3×10^8	80.34	60.7
VGG16	Real	1.6×10^8	74.88	62.87
	RH-emo+Quat	1×10^7	72.25	71.10
AlexNet	Real	5.7×10^7	71.02	63.33
	RH-emo+Quat	1×10^7	71.81	70.31
ResNet	Real	2.3×10^7	61.05	57.20
	RH-emo+Quat	4.9×10^6	73.03	**71.20**

TABLE II
RESULTS FOR RAVDESS

Arch.	Method	Params	Train acc.	Test acc.	Test UAR
VGG16	Real	1.6×10^8	47.10	41.06	40.07
	RH-emo+Quat	1×10^7	55.50	49.85	**48.28**
	Real-Pre	1.6×10^8	67.86	45.30	46.33
	RH-emo+Quat-Pre	1×10^7	67.08	53.79	46.88
AlexNet	Real	5.7×10^7	54.55	46.36	36.36
	RH-emo+Quat	1×10^7	50.62	43.94	38.21
	Real-Pre	5.7×10^7	83.54	51.06	45.71
	RH-emo+Quat-Pre	1.4×10^7	63.16	47.58	41.29
ResNet	Real	2.3×10^7	72.84	43.48	33.16
	RH-emo+Quat	4.9×10^6	91.29	**55.15**	46.51
	Real-Pre	2.3×10^7	22.16	18.79	13.33
	RH-emo+Quat-Pre	4.9×10^6	89.54	52.42	44.27

TABLE III
RESULTS FOR EMODB

Arch.	Method	Params	Train acc.	Test acc.	Test UAR
VGG16	Real	1.6×10^8	72.74	70.00	58.86
	RH-emo+Quat	1×10^7	79.54	50.00	41.73
	Real-Pre	1.6×10^8	78.16	52.00	46.95
	RH-emo+Quat-Pre	1×10^7	75.00	47.00	40.11
AlexNet	Real	5.7×10^7	63.1	47.00	40.77
	RH-emo+Quat	1×10^7	82.3	49.00	41.99
	Real-Pre	5.7×10^7	71.45	67.00	59.93
	RH-emo+Quat-Pre	1.4×10^7	77.63	71.00	63.89
ResNet	Real	2.3×10^7	99.47	48.00	42.76
	RH-emo+Quat	4.9×10^6	99.73	**73.00**	**65.64**
	Real-Pre	2.3×10^7	100.00	72.00	64.04
	RH-emo+Quat-Pre	4.9×10^6	99.73	46.00	38.34

layer between the convolution-based feature extractor and the fully-connected classifier. This permits to obtain an identical output shape from the feature extractor for any input dimension. We removed this layer from only VGG16, in order to test the behavior of our approach also in this situation. Doing this, in fact, the feature extractor presents a reduced output dimensionality when the networks are fed with the quaternion embeddings (75% smaller than using the real spectrograms), enabling to spare of a major number of network parameters.

For all experiments we used a learning rate of 0.00001, ADAM optimizer, and a batch size of 20 samples, we apply early stopping with the patience of 20 epochs on the validation loss and we split the training, validation, and test sub-sets with approximately 70%, 20% and 10% of the data, respectively.

The main aim of this research is to provide a valid comparison between the proposed approach (quaternion-valued CNNs fed with RH-Emo embeddings) and standard equivalent real-valued

TABLE IV
RESULTS FOR TESS

Arch.	Method	Params	Train acc.	Test acc.	Test UAR
VGG16	Real	1.6×10^8	99.54	97.62	98.51
	RH-emo+Quat	1×10^7	98.87	97.62	96.67
	Real-Pre	1.6×10^8	99.95	99.52	98.95
	RH-emo+Quat-Pre	1×10^7	98.72	97.85	97.92
AlexNet	Real	5.7×10^7	99.18	98.01	97.03
	RH-emo+Quat	1×10^7	99.54	98.56	97.34
	Real-Pre	5.7×10^7	100.00	98.01	98.95
	RH-emo+Quat-Pre	1.4×10^7	99.75	98.81	97.38
ResNet	Real	2.3×10^7	100.00	97.38	97.84
	RH-emo+Quat	4.9×10^6	100.00	**99.76**	**99.58**
	Real-Pre	2.3×10^7	59.88	57.53	56.72
	RH-emo+Quat-Pre	4.9×10^6	100.00	99.28	97.91

TABLE V
TEST ACCURACY RESULTS

Dataset	Average improvement			Best improvement
	No pret.	Pret.	Overall	
IEMOCAP	9.74	/	/	7.87
RAVDESS	6.01	12.88	9.45	4.09
EmoDB	2.34	-9.00	-3.34	1.00
TESS	0.97	13.63	7.30	0.24

architectures, isolating as much as possible the pure difference between them. We configured our experimental setup in order to show the performance difference between real and corresponding quaternion CNNs fed with the emotional quaternion embeddings. Therefore, we paid attention to performing each experiment in as-close-as-possible conditions, rather than optimizing each architecture for each different dataset, in order to highlight the properties of our approach. State-of-the-art results for SER tasks usually involve more complex solutions, as, among others, data augmentation [66], [67], [68], [69], attention [66], [69], [70], [71], adversarial attacks [72], multimodal processing [70], [73], speaker-aware processing [74], [75], transformer designs [70], [75]. Moreover, the state-of-the-art approach can be radically different for each dataset, and therefore using the best method for each dataset would not permit having the same configuration for all possible aspects in both RH-Emo experiments and the baselines. This would add much more complexity to the setup, consequently making it less straightforward to isolate and understand the properties of our approach.

Because of these reasons and the fact that many existing studies are based on different methods to compute the scores, different data splits and may use multiple data domains, our results can not be directly compared to the current state-of-the-art accuracy for these datasets, which, to the best of our knowledge are 75.60% for IEMOCAP [71], 87.5% for RAVDESS [73], 88.47% for EmoDb [66] and 99.6% for TESS [67].

B. Experimental Results

Table I shows the pretraining results we obtained on IEMOCAP, while Tables II, III, and IV provide the results on RAVDESS, EmoDB, and TESS, respectively. Table V, shows

the average and best test accuracy improvement provided by our approach, among all CNN architectures for each dataset. Here, average improvement refers to the difference between the average test accuracy among all real-valued and all quaternion-valued outcomes, whereas the best improvement is the difference between the best real-valued and the best quaternion-valued accuracy we obtained. For the core results (Tables II, III, and IV) we include also the test set results in terms of Unweighted Average Recall (UAR). This gives further insight into the model's generalization performance with a metric that does not take into account possible imbalance of the datasets' labels.

The results clearly show that our approach enhances the model's performance while improving its efficiency. For all datasets, the quaternion CNNs fed with RH-emo embeddings provide the best test accuracy overall, with an accuracy improvement of 6.01 percentage points (pp) for RAVDESS, 2.34 pp for EmoDB, and 0.97 for TESS in the case we do not apply CNN pretraining. The only case where our approach does not improve the test accuracy is with the EmoDB dataset, applying CNN pretraining, where we have a performance drop of 9 pp. In the other cases where we applied CNN pretraining, our approach provides a strong average improvement of 12.88 and 13.63 pp, respectively for RAVDESS and TESS. Moreover, the test set results in terms of UAR metric confirm the overall trend of the accuracy metric. Nevertheless, in one single case (VGG-16 network on RAVDESS) there is a narrow inconsistency between the two metrics. Here the pretrained QCNN shows the best test accuracy, while the best UAR score is given by the non-pretrained QCNN.

The results computed on IEMOCAP (Table I and first row of Table V) depict a limit case, where knowledge is not transferred to different data because the same dataset is used for the RH-emo pretraining and for SER. Therefore here we did not apply any CNN pretraining. Also in this special case is evident that models benefit from the use of quaternion-valued SER CNNs fed with emotional embeddings, with an average improvement of 9.74 pp among all CNN designs we tested.

VI. Ablation Studies

In order to further explore the properties of our approach and to support its foundations, we performed additional experiments and ablation studies. For these studies we applied the same experimental setup presented in Section V, altering only specific details, as described below.

A. Removing RH-emo Components

In this study, we alter the RH-emo structure and test the emotion recognition accuracy using the embeddings generated from the modified RH-emo networks. We compared the full RH-emo, as described in Section IV, to the following altered versions:

- Real: identical to the regular network, but the decoder part is real-valued and no split activation is applied to the reconstructed output in the loss function.
- Reconstruction only: we removed the supervised classification branch, resulting in a completely unsupervised real-quaternion hybrid autoencoder.
- Emotion only: we removed the unsupervised reconstruction branch from the network, obtaining a completely supervised and real-valued emotion classification CNN. In this configuration, there are still 4 target outputs, each with a dedicated classifier (discrete emotion, valence, arousal, dominance).
- Discrete emotion only: we removed the valence, arousal, and dominance classifiers, keeping only the discrete emotion classification branch. The rest of the network is unaltered.
- valence-arousal-dominance only: we removed the discrete emotion recognition branch, keeping only the branches for valence, arousal, and dominance. The rest of the network is unaltered.

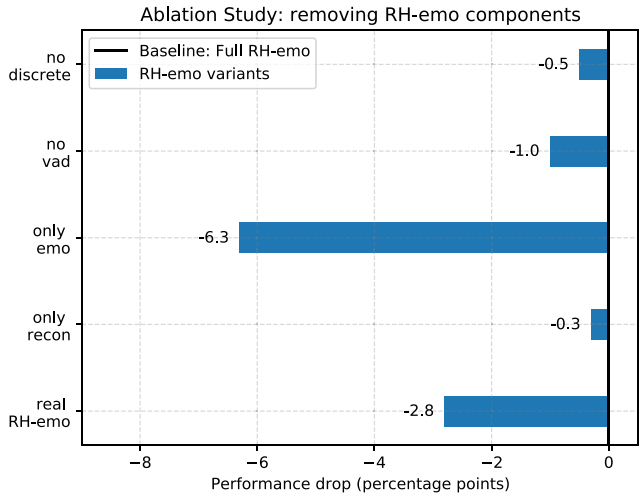

Fig. 3. Ablation study results. The x axis shows the average drop in test accuracy (among the quaternion-valued VGG16, AlexNet and ResNet-50 for all corpora) obtained with different variants of RH-emo. Each row refers to a variant of RH-emo where we removed a specific component, namely: a completely real-valued network, only reconstruction, only emotion recognition, no valence-arousal-dominance (vad) estimation, and no discrete emotion classification.

Fig. 3 exposes the results of this ablation study. In the figure, we show the mean test accuracy improvement obtained for all corpora with the quaternion-valued VGG16, AlexNet, and ResNet-50 over the real-valued baselines. Each row shows the results obtained feeding the quaternion-valued networks with the embeddings created with the above-described variants of RH-emo. These results consistently confirm the foundation of our approach. The performance of all variants is inferior to the full RH-emo. In addition, we recall that the quaternion-valued CNNs fed with the emotional embeddings use a considerably lower amount of parameters. The results point out that the unsupervised branch of RH-emo is fundamental to obtain useful embeddings, in fact, the emotion-only version, where the decoder part of RH-emo is removed, provides the most severe drop in performance compared to all variants and also the baseline. As

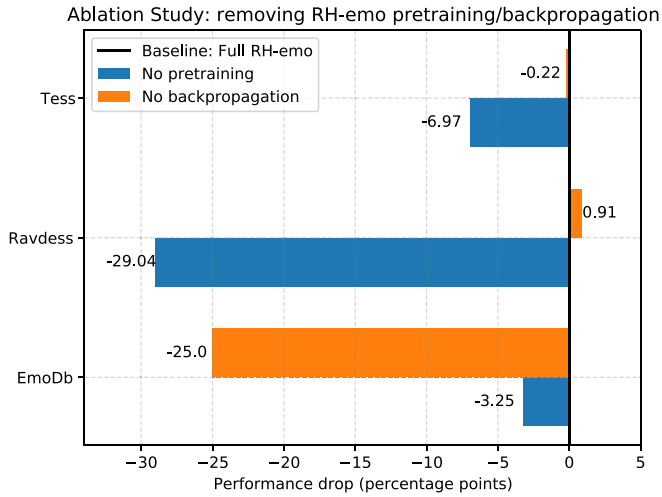

Fig. 4. Ablation study results. The x axis shows the average difference in test accuracy (among the quaternion-valued VGG16, AlexNet and ResNet-50) obtained by removing the RH-emo pretraining (blue lines) and backpropagation (orange lines).

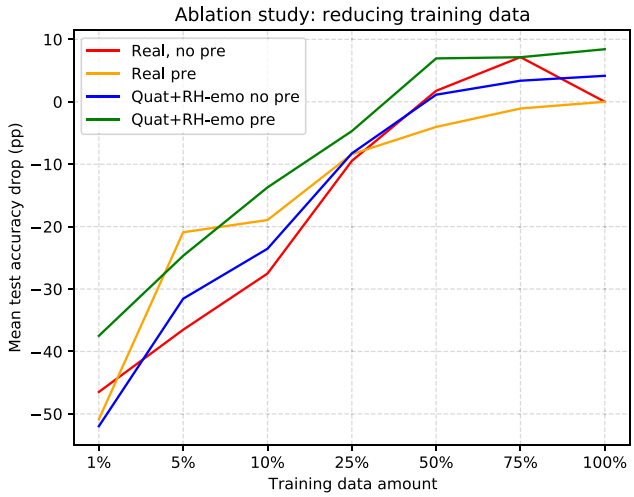

Fig. 5. Ablation study results. The y axis shows the test accuracy drop of each model, compared to the baselines that use 100% of the training data. Each point in the line shows the average performance among the real-valued (red, yellow) and quaternion-valued (blue, green) VGG16, AlexNet, and ResNet-50 architectures for all corpora. The x axis shows the percentage of available training and validation data used. The data reduction rates shown in the x axis are a discrete set: we trained only on the data percentage values that are shown and not on intermediate values. We use the full test set in all cases, in order to have a consistent performance measure also with.

we expected, the quaternion-valued decoder of the actual RH-emo outperforms the completely real-valued version (by 2.8pp). This supports our hypothesis that a quaternion-value decoder is able to create embeddings that present more suitable intra-channel correlations for the quaternion-valued CNNs. Moreover, also here, the quaternion approach leads to faster (pre)training and less memory demand due to the lower amount of parameters. The completely unsupervised variant (recognition-only) is conceptually similar to R2Hae [55], but it relies on a convolutional design and it is applied to a different domain. This ablation study shows that the addition of a classification branch to R2Hae provides an improvement in performance (by 0.3 pp in our case) and therefore the semi-supervision can be considered a valuable extension to R2Hae. This ablation study also shows that the classification of emotion in the valence-arousal-dominance space is more influential in the creation of stronger embeddings. In fact, the RH-emo variant without discrete classification provides superior accuracy compared to the discrete-only version (by 0.5 pp) This is further supported by the fact that, as a result of an extensive grid search, we apply a stronger weight to the valence-arousal-dominance term of the loss function (the α term in (4)).

B. Removing RH-Emo Pretraining and Backpropagation

We performed an additional ablation study where we alter how the RH-emo weights are initialized and backpropagated during the SER training. Fig. 4 depicts the results of this study, showing the average difference in test accuracy per-dataset among all CNN designs. On the one hand, we initialized the weights of RH-emo with random values while we regularly backpropagate the gradients of the RH-emo's encoder layers (blue rows). By doing this, we completely ignore the RH-emo pretraining and we force the QCNN network to perform end-to-end training, directly learning how to map the real-valued input spectrograms into quaternion-compatible representations to feed the QCNNs. This approach is conceptually similar to (R2He) [55]. On the other hand, we regularly initialize the weights of RH-emo with the pretrained RH-emo network, but we don't backpropagate the RH-emo layers (orange rows). The results of this experiment strongly support the foundation of our approach. The removal of RH-emo pretraining causes a consistent and substantial decrease in the QCNNs test performance, of 29.4, 3.25, and 6.97 pp for RAVDESS, EmoDB, and TESS, respectively. This confirms the importance of the prior training of the RH-emo encoder, as exposed in Section IV, for the development of adequate quaternion emotional embeddings. On the contrary, the lack of backpropagation of the RH-emo layers does not provide a consistent performance drop. While the performance decreases for EmoDB (25 pp) and for TESS (0.22 pp), a narrow accuracy boost is evident for RAVDESS (0.91 pp). Moreover, the performance difference is averagely inferior compared to the no-pretraining case.

C. Reducing Training Data

As a further study, we re-trained all CNNs and QCNNs, progressively decreasing the amount of training and validation data. The size of the test set, instead, is kept unaltered, in order to have a consistent performance measure that can be compared with the other results presented in this paper. Fig. 5 shows the outcomes of this experiment. Each line shows the trend of the average test accuracy among all CNN architectures, at different reduction rates of the data. Specifically, we trained on 100%, 75%, 50%, 25%, 10%, 5% and 1% of the available data. The yellow and red lines are the baselines, respectively with and without CNN pretraining on IEMOCAP. Instead, the green and

blue lines show the trend for the QCNNs + RH-emo, respectively with and without CNN pretraining.

The results of this ablation study clearly point out that our method can provide consistent performance improvement even in conditions with less data. In all cases but one (5% of training data) our pretrained approach surpasses both real-valued baselines. This is a convenient property for SER tasks, considering the general scarcity of emotion-labelled speech audio data.

VII. Discussion

A. Resource Savings

RH-emo permits to spare of a considerable amount of parameters. Compared to the real counterparts, the quaternion VGG16 uses the ∼6% of the parameters, while the quaternion AlexNet and ResNet-50 use the ∼25%. The difference between the VGG16 and the others is due to the lack of adaptive average pooling (as described above). Therefore, on the one hand, the use of quaternion-valued layers instead of real-valued ones permits to drop in the number of parameters by a factor of 0.25, while, on the other hand, the smaller feature dimensionality obtained with the embeddings further cuts down the number of parameters by a factor of 0.25. This in turn permits the reduction of the model's memory requirements and training time. In our implementation, the embedding computation happens during the training for every batch and, therefore, both the main network and the RH-emo feature extractor are loaded into the memory. This simulates a plausible application scenario of RH-emo, where the embeddings need to be computed in real-time. Although it is possible to pre-compute the embeddings as part of the pre-processing pipeline, further reducing the memory demand and computation time. As regards the memory demand, in our setup the quaternion networks require on average 84.2% of memory, compared to their real-valued equivalents. For the VGG16 (where we don't apply average pooling) the memory demand is approximately 70%, for AlexNet the 89%, and for ResNet-50 the 93%. Regarding the training time, the epoch duration of our quaternion networks compared to the real networks is approximately 15.9% for VGG16, 88.1% for AlexNet, and 162.6% for ResNet-50. These outcomes show that the maximum efficiency in terms of both memory demand and computation time is obtained for VGG16, where we take advantage of the reduced dimensionality of the embeddings. On the other hand, the accuracy improvement for ResNet-50 comes at the cost of an increased computation time with respect to the real networks, but still reducing the model's memory demand.

B. Reconstruction Properties

Fig. 6 shows an example of the decoder's output of the pretrained RH-emo model. The *Input* subplot is the input magnitudes-only spectrogram and the *Output: mean* is the element-wise mean of the quaternion separate axes and, therefore, the actual matrix that is compared to the input in the loss function. The sub-plots labelled as *Output: real*, $\hat{\imath}, \hat{\jmath}, \hat{\kappa}$ depict

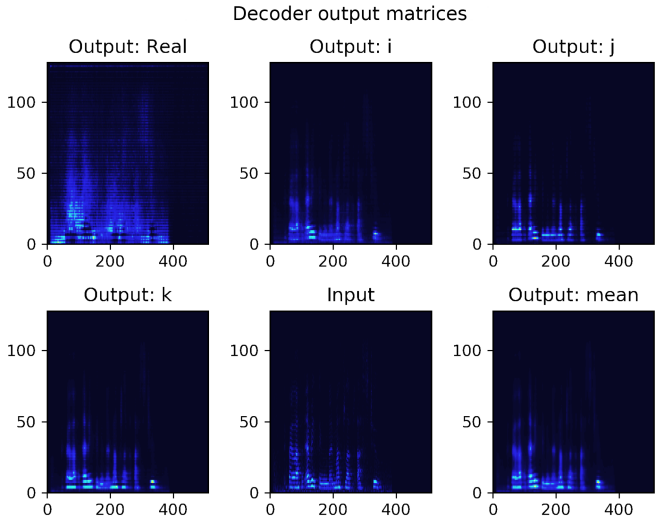

Fig. 6. Example of RH-emo quaternion reconstruction of a speech spectrogram. *Input* is the magnitudes-only input spectrogram, *Output: real*, $\hat{\imath}, \hat{\jmath}, \hat{\kappa}$ are the four output matrices of RH-emo, respectively reconstructed from the discrete emotion, valence, arousal and dominance axes of the embeddings, *Output: mean* is the pixel-wise average of *Output: real*, $\hat{\imath}, \hat{\jmath}, \hat{\kappa}$ and is the matrix that is compared to the input in the loss function.

the separate quaternion axes, which are generated from the emotional embeddings: *real* from the discrete emotion classification matrix, and $\hat{\imath}, \hat{\jmath}, \hat{\kappa}$ from the valence, arousal, and dominance channels, respectively.

By comparing the *Input* and the *Output: mean* plots, it is evident that the reconstruction is not perfect. While the time-wise articulation of the speech seems to be accurately reproduced, the model is not able to reconstruct in detail the most feeble harmonics of the signal. Although it is interesting the way the different quaternion axes are differentiated. In the real axis, the model seems to perform an operation similar to amplitude compression (obtainable, for instance, by computing the square root of the matrix), bringing up the signal's quietest portions around the speech region. Instead, in the 3 complex axes ($\hat{\imath}, \hat{\jmath}, \hat{\kappa}$) different aspects of the signal are highlighted, focusing on different harmonics and/or temporal areas. Our intuition is that these representations may represent different "emotional points of view" of the input speech signal.

C. Limitations

Besides the numerous advantages that our approach provides, there are also some intrinsic limitations. The main constraint of our approach is that a pretrained RH-emo network can be used for only a fixed time scale. In this paper, we considered a temporal window of 4 seconds, which is well suited for most SER tasks and datasets. If a different time scale is needed, then a specific RH-emo has to be trained on purpose. Another limitation is that training with an end-to-end fashion is not possible, as a pre-trained RH-emo is needed and the omission of the RH-emo pretraining stage leads to a drastic decrease in the model's performance, as shown in Section VI-B.

D. Applications and Future Work

The advantages provided by the combination of RH-emo and quaternion-valued networks suggest several application scenarios. Due to the substantial saving of trainable parameters, memory, and training time, our approach is particularly suited for situations where limited resources are available and performance can not be sacrificed. Another useful property of RH-emo is that while the embeddings carry the necessary information to perform SER tasks (as proven by our experimental results), they also provide speaker anonymity, as it is not possible to reconstruct the input spectrogram without the RH-emo pretrained weights. This could be exploited in situations where sensible speech data must be used for SER tasks.

The positive results we obtained justify further investigation of this approach. An immediate research objective is to test RH-emo with different datasets, and architectures (including recurrent networks), with multiple time scales and to different tasks. In particular, we intend to apply the same principle of RH-emo (based on a semi-supervised autoencoder where each embedded channel is optimized for the classification of a different characteristic of an entity) for different tasks, where a quadral representation of input data can not be directly inferred from data, as for speech emotion. An example of this is music genre recognition tasks, where the embedded dimensions of the autoencoder are optimized for tempo, harmonic key, spoken words, and instrument type recognition.

VIII. Conclusion

In this paper we presented RH-emo, a semi-supervised approach to obtain quaternion emotional embeddings from real speech spectrograms. This method enables to perform speech emotion recognition tasks with quaternion-valued convolutional neural networks, using real-valued magnitudes spectrograms as input. We use RH-emo pretrained on IEMOCAP to extract quaternion embeddings from speech spectrograms, where the individual axes are optimized for the classification of different emotional characteristics: valence, arousal, dominance, and overall discrete emotion.

We compare the performance on SER tasks of real-valued CNNs fed with regular spectrograms and quaternion-valued CNNs fed with RH-emo embeddings. We evaluate our approach on a variety of cases, using 4 popular SER datasets (IEMOCAP, RAVDESS, EmoDB, TESS) and with 3 widely-used CNN designs of increasing capacity (ResNet-50, AlexNet and VGG16). Our approach provides a consistent improvement in the test accuracy for all datasets while using a considerably lower amount of resources. We obtained an average improvement of 6.01 pp for RAVDESS, 2.34 pp for EmoDB, and 0.97 pp for TESS and we spared up to 94% of the trainable parameters, up to the 30% of GPU memory and up to 84.1% of training time. Moreover, we performed additional experiments and ablations studies that confirm the properties and foundations of our approach. The results show that the combination of RH-emo and QCNNs is a suitable strategy to circumvent the high resource demand of SER models and that our approach provides consistent performance improvement also in scenarios where the available training data is scarce.

The positive results justify further investigation of this approach. An immediate research objective is to test RH-emo with different datasets, architectures (including recurrent networks), with multiple signal dimensions, and different tasks.

Acknowledgment

The authors would like to thank the NVIDIA Applied Research Accelerator Program for the donation of an NVIDIA Quadro RTX 8000 for the project “Quaternion Deep Learning for 3D Audio Sources”.

References

[1] J. Rybka and A. Janicki, “Comparison of speaker dependent and speaker independent emotion recognition,” *Int. J. Appl. Math. Comput. Sci.*, vol. 23, no. 4, pp. 797–808, 2013.

[2] V. Hozjan and Z. Kačič, “Context-independent multilingual emotion recognition from speech signals,” *Int. J. Speech Techn.*, vol. 6, no. 3, pp. 311–320, 2003.

[3] S. Rigoulot, E. Wassiliwizky, and M. D. Pell, “Feeling backwards? How temporal order in speech affects the time course of vocal emotion recognition,” *Front. Psychol.*, vol. 4, no. 367, pp. 1–14, 2013.

[4] S. Khorram, Z. Aldeneh, D. Dimitriadis, M. McInnis, and E. M. Provost, “Capturing long-term temporal dependencies with convolutional networks for continuous emotion recognition,” in *Proc. INTERSPEECH*, 2017, pp. 1253–1257.

[5] Z. Lian, J. Tao, B. Liu, and J. Huang, “Unsupervised representation learning with future observation prediction for speech emotion recognition,” in *Proc. INTERSPEECH*, 2019, pp. 3840–3844.

[6] G. K. Verma and U. S. Tiwary, “Affect representation and recognition in 3 D continuous valence–arousal–dominance space,” *Multimedia Tools Appl.*, vol. 76, no. 2, pp. 2159–2183, 2017.

[7] M. Gaertner, D. Sauter, H. Baumgartl, T. Rieg, and R. Buettner, “Multi-class emotion recognition within the valence-arousal-dominance space using EEG,” in *Proc. AMCIS*, 2021.

[8] S. Buechel and U. Hahn, “Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis,” in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, Valencia, Spain, vol. 2, Short Papers, pp. 578–585, Apr. 2017.

[9] M. W. Bhatti, Y. Wang, and L. Guan, “A neural network approach for human emotion recognition in speech,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Vancouver, BC, vol. 2, 2004, pp. 181–184.

[10] R. Cowie et al., “Emotion recognition in human-computer interaction,” *IEEE Signal Process. Mag.*, vol. 18, no. 1, pp. 32–80, Jan. 2001.

[11] J. Nicholson, K. Takahashi, and R. Nakatsu, “Emotion recognition in speech using neural networks,” *Neural Comput. Appl.*, vol. 9, no. 4, pp. 290–296, 2000.

[12] D. Ververidis and C. Kotropoulos, “Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition,” *Signal Process.*, vol. 88, no. 12, pp. 2956–2970, 2008.

[13] X. Mao, L. Chen, and L. Fu, “Multi-level speech emotion recognition based on HMM and ANN,” in *Proc. WRI World Congr. Comput. Sci. Inf. Eng.*, 2009, vol. 7, pp. 225–229.

[14] T. L. Nwe, S. W. Foo, and L. C. D. Silva, “Speech emotion recognition using hidden Markov models,” *Speech Commun.*, vol. 41, no. 4, pp. 603–623, 2003.

[15] J. Zhou, G. Wang, Y. Yang, and P. Chen, “Speech emotion recognition based on rough set and SVM,” in *Proc. IEEE Int. Conf. Cogn. Inform.*, 2006, vol. 1, pp. 53–61.

[16] H. Hu, M.-X. Xu, and W. Wu, “GMM supervector based SVM with spectral features for speech emotion recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Honolulu, HI, USA, vol. 4, 2007, pp. 413–416.

[17] D. Neiberg, K. Elenius, and K. Laskowski, “Emotion recognition in spontaneous speech using GMMs,” in *Interspeech*., Pittsburgh, PA, Sep. 2006, pp. 809–812.

[18] M. El Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” *Pattern Recognit.*, vol. 44, no. 3, pp. 572–587, 2011.

[19] A. M. Badshah, J. Ahmad, N. Rahim, and S. W. Baik, "Speech emotion recognition from spectrograms with deep convolutional neural network," in *Proc. IEEE Int. Conf. Platform Technol. Serv.*, 2017, pp. 1–5.

[20] T.-W. Sun, "End-to-end speech emotion recognition with gender information," *IEEE Access*, vol. 8, pp. 152423–152438, 2020.

[21] D. Issa, M. F. Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," *Biomed. Signal Process. Control*, vol. 59, pp. 1–11, 2020.

[22] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," in *Proc. INTERSPEECH*, 2015, pp. 1537–1540.

[23] V. Chernykh and P. Prikhodko, "Emotion recognition from speech with recurrent neural networks," Jul. 2018, *arXiv:1701.08071v2*.

[24] G. Trigeorgis et al., "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 5200–5204.

[25] P. Meyer, Z. Xu, and T. Fingscheidt, "Improving convolutional recurrent neural networks for speech emotion recognition," in *Proc. IEEE Spoken Lang. Technnol. Workshop*, 2021, pp. 365–372.

[26] M. A. Qamhan, A. H. Meftah, S.-A. Selouani, Y. A. Alotaibi, M. Zakariah, and Y. M. Seddiq, "Speech emotion recognition using convolutional recurrent neural networks and spectrograms," in *Proc. IEEE Can. Conf. Elect. Comput. Eng.*, 2020, pp. 1–5.

[27] Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 3687–3691.

[28] Q. Mao, M. Dong, Z. Huang, and Y. Zhan, "Learning salient features for speech emotion recognition using convolutional neural networks," *IEEE Trans. Multimedia*, vol. 16, pp. 2203–2213, 2014.

[29] Z. Huang, M. Dong, Q. Mao, and Y. Zhan, "Speech emotion recognition using CNN," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 801–804.

[30] K. Han, D. Yu, and I. Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," in *Proc. INTERSPEECH*, 2014, pp. 223–227.

[31] N. Rossenbach, A. Zeyer, R. Schlüter, and H. Ney, "Generating synthetic audio data for attention-based speech recognition systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7069–7073.

[32] A. Laptev, R. Korostik, A. Svischev, A. Andrusenko, I. Medennikov, and S. Rybin, "You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation," in *Proc. Int. Congr. Image Signal Process., Biomed. Eng. Informat.*, 2020, pp. 439–444.

[33] M. Macary, M. Tahon, Y. Estève, and A. Rousseau, "On the use of self-supervised pre-trained acoustic and linguistic features for continuous speech emotion recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 373–380.

[34] L. Pepino, P. Riera, and L. Ferrer, "Emotion recognition from speech using Wav2vec 2.0 embeddings," in *Proc. INTERSPEECH*, 2021, pp. 3400–3404.

[35] E. Guizzo, T. Weyde, and G. Tarroni, "Anti-transfer learning for task invariance in convolutional neural networks for speech processing," *Neural Netw.*, vol. 142, pp. 238–251, 2021.

[36] S. Padi, D. Manocha, and R. D. Sriram, "Multi-window data augmentation approach for speech emotion recognition," Feb. 2022, *arXiv:2010.09895v4*.

[37] A. Shilandari, H. Marvi, and H. Khosravi, "Speech emotion recognition using data augmentation method by cycle-generative adversarial networks," *Signal, Image Video Process*., vol. 16, no. 7, pp. 1955–1962, 2022.

[38] P. Fewzee and F. Karray, "Dimensionality reduction for emotional speech recognition," in *Proc. IEEE Int. Conf. Privacy, Secur., Risk Trust Int. Conf. Social Comput.*, 2012, pp. 532–537.

[39] N. Patel, S. Patel, and S. H. Mankad, "Impact of autoencoder based compact representation on emotion detection from audio," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 2, pp. 867–885, 2021.

[40] Y. Tay et al., "Lightweight and efficient neural natural language processing with quaternion networks," in *Proc. 57th Ann. Meeting the Assoc. Comput. Linguistics*, 2019, pp. 1494–1503.

[41] E. Grassucci, D. Comminiello, and A. Uncini, "A quaternion-valued variational autoencoder," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 3310–3314.

[42] E. Grassucci, E. Cicero, and D. Comminiello, "Quaternion generative adversarial networks," in *Generative Adversarial Learning: Architectures and Applications*, R. Razavi-Far, A. Ruiz-Garcia, V. Palade, and J. Schmidhuber, Eds. Cham, Switzerland:Springer, 2022, pp. 57–86.

[43] E. Grassucci, A. Zhang, and D. Comminiello, "PHNNs: Lightweight neural networks via parameterized hypercomplex convolutions," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 2022, doi: 10.1109/TNNLS.2022.3226772.

[44] A. B. Greenblatt and S. S. Agaian, "Introducing quaternion multi-valued neural networks with numerical examples," *Inf. Sci.*, vol. 423, pp. 326–342, 2018.

[45] T. Parcollet et al., "Quaternion convolutional neural networks for end-to-end automatic speech recognition," in *Proc. INTERSPEECH*, 2018, pp. 22–26.

[46] A. Muppidi and M. Radfar, "Speech emotion recognition using quaternion convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 6309–6313.

[47] T. Bulow and G. Sommer, "Hypercomplex signals-a novel extension of the analytic signal to the multidimensional case," *IEEE Trans. Signal Process.*, vol. 49, no. 11, pp. 2844–2852, Nov. 2001.

[48] D. P. Mandic, C. Jahanchahi, and C. C. Took, "A quaternion gradient operator and its applications," *IEEE Signal Process. Lett.*, vol. 18, no. 1, pp. 47–50, Jan. 2011.

[49] D. Comminiello, M. Scarpiniti, R. Parisi, and A. Uncini, "Frequency-domain adaptive filtering: From real to hypercomplex signal processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 7745–7749.

[50] D. Comminiello, M. Lella, S. Scardapane, and A. Uncini, "Quaternion convolutional neural networks for detection and localization of 3D sound events," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 8533–8537.

[51] R. K. Furness, "Ambisonics–an overview," in *Proc. AES 8th Int. Conf. Sound Audio. Audio Eng. Soc.*, pp. 181–190, Washington, DC, May 1990.

[52] X. Qiu, T. Parcollet, M. Ravanelli, N. Lane, and M. Morchid, "Quaternion neural networks for multi-channel distant speech recognition," in *Proc. INTERSPEECH*, 2020, pp. 329–333.

[53] C. Brignone, G. Mancini, E. Grassucci, A. Uncini, and D. Comminiello, "Efficient sound event localization and detection in the quaternion domain," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 69, no. 5, pp. 2453–2457, May 2022.

[54] E. Grassucci, G. Mancini, C. Brignone, A. Uncini, and D. Comminiello, "Dual quaternion ambisonics array for six-degree-of-freedom acoustic representation," *Pattern Recognit. Lett.*, vol. 166, pp. 24–30, 2023.

[55] T. Parcollet, M. Morchid, X. Bost, G. Linarès, and R. D. Mori, "Real to H-space autoencoders for theme identification in telephone conversations," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 198–210, 2020.

[56] C. J. Gaudet and A. S. Maida, "Deep quaternion networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2018, pp. 1–8.

[57] B. C. Ujang, C. C. Took, and D. P. Mandic, "Quaternion-valued nonlinear adaptive filtering," *IEEE Trans. Neural Netw.*, vol. 22, no. 8, pp. 1193–1206, Aug. 2011.

[58] C. Busso et al., "IEMOCAP: Interactive emotional dyadic motion capture database," *Lang. Resour. Eval.*, vol. 42, no. 4, pp. 335–359, 2008.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, San Diego, CA, May 2015, pp. 1–13.

[60] S. R. Livingstone and F. A. Russo, "The Ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PLoS One*, vol. 13, no. 5, 2018, Art. no. e0196391.

[61] F. Burkhardt, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *Proc. Eur. Conf. Speech Commun. Technol*., 2005, pp. 1517–1520.

[62] K. Dupuis and M. K. Pichora-Fuller, "Recognition of emotional speech for younger and older talkers: Behavioural findings from the toronto emotional speech set," *Can. Acoust.*, vol. 39, no. 3, pp. 182–183, 2011.

[63] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, San Diego, CA, May 2015, pp. 1–14.

[64] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2016, pp. 770–778.

[66] N. T. Pham, D. N. M. Dang, and S. D. Nguyen, "Hybrid data augmentation and deep attention-based dilated convolutional-recurrent neural networks for speech emotion recognition," 2021, *arXiv:2109.09026*.

[67] S. Jothimani and K. Premalatha, "MFF-SAUG: Multi feature fusion with spectrogram augmentation of speech emotion recognition using convolution neural network," *Chaos, Solitons Fractals*, vol. 162, 2022, Art. no. 112512.
[68] C. Etienne, G. Fidanza, A. Petrovskii, L. Devillers, and B. Schmauch, "Speech emotion recognition with data augmentation and layer-wise learning rate adjustment," Feb. 2018 *arXiv:1802.05630*.
[69] M. Xu, F. Zhang, X. Cui, and W. Zhang, "Speech emotion recognition with multiscale area attention and data augmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 6319–6323.
[70] N.-H. Ho, H.-J. Yang, S.-H. Kim, and G. Lee, "Multimodal approach of speech emotion recognition using multi-level multi-head fusion attention-based recurrent neural network," *IEEE Access*, vol. 8, pp. 61672–61686, 2020.
[71] S. Kakouros, T. Stafylakis, L. Mosner, and L. Burget, "Speech-based emotion recognition with self-supervised models using attentive channel-wise correlations and label smoothing," in *Proc. 48th IEEE Int. Conf. Acoust., Speech, Signal Process.*, Rhodes, Greece, Jun. 2023.
[72] S. Latif, R. Rana, and J. Qadir, "Adversarial machine learning and speech emotion recognition: Utilizing generative adversarial networks for robustness," 2018, *arXiv:1811.11402*.
[73] Y. L. Bouali, O. B. Ahmed, and S. Mazouzi, "Cross-modal learning for audio-visual emotion recognition in acted speech," in *Proc. IEEE 6th Int. Conf. Adv. Technol. Signal Image Process.*, 2022, pp. 1–6.
[74] T. Kim and P. Vossen, "Emoberta: Speaker-aware emotion recognition in conversation with RoBERTa," 2021, *arXiv:2108.12009*.
[75] J. Li, D. Ji, F. Li, M. Zhang, and Y. Liu, "Hitrans: A transformer-based context-and speaker-sensitive model for emotion detection in conversations," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 4190–4200.

Eric Guizzo received the Master degree in sound and music computing from the Conservatorio Pollini of Padua, Padua, Italy. He is currently working toward the Ph.D. degree with the Computer Science Department of City University of London, London, U.K. He has completed a research fellowship with the DIET Department of Sapienza University of Rome, Rome, Italy. His academic research is focused on audio applications of machine learning. He has primarily worked on methods to enhance the performance of Speech Emotion Recognition models and the independence of their predictions from unwanted biases. On the other hand, he is exploring the potential of deep learning in creative contexts related to Sound Art. This engages him in constant experimentation, applying novel paradigms of human-machine interaction and intelligent sound synthesis.

Tillman Weyde (Member, IEEE) Ph.D., is currently a Reader with the Department of Computer Science, City, University of London, London, U.K., the Head of the Machine Intelligence and Media Informatics Research Group, and a Member of the Machine Learning Research Centre and the Data Science Institute. He has been working in the field of machine learning and artificial intelligence for more than 25 years and has authored or coauthored more than 150 peer-reviewed papers. Prior to joining City in 2005, he was a Postdoctoral Researcher with the University of Osnabrück, Osnabrück, Germany. His research interests include developing new methods for machine learning, especially using prior knowledge, and on AI applications. He has run several research projects funded by the EU, EPSRC, ARHC, NEH, and others. Tillman is a Member of the EPSRC College, and he has won multiple awards for published papers and software. In addition to academic activities, he works with start-ups to develop AI applications in different industries.

Simone Scardapane received the Ph.D. degree in information and communication technologies from the Sapienza University of Rome, Rome, Italy, in 2016. He is a tenure-track Assistant Professor with Sapienza University of Rome. He has authored or coauthored more than 100 papers on these topics in top-tier journals and conferences. His research interests include graph neural networks, explainability, continual learning and, more recently, modular and efficient deep networks. He is currently an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (IEEE), *Neural Networks* (Elsevier), *Industrial Artificial Intelligence* (Springer), and *Cognitive Computation* (Springer). He is a Member of multiple groups and societies, including the ELLIS society, the IEEE Task Force on Reservoir Computing, the Machine learning in geodesy Joint Study Group of the International Association of Geodesy, and the Statistical Pattern Recognition Techniques TC of the International Association for Pattern Recognition.

Danilo Comminiello (Senior Member, IEEE) received the Laurea degree in telecommunication engineering and the Ph.D. degree in information and communication engineering from the Sapienza University of Rome, Italy, in 2008 and 2012, respectively. He is currently an Associate Professor with the Department of Information Engineering, Electronics and Telecommunications (DIET), Sapienza University of Rome, Italy. His research interests include design and analysis of neural networks and machine learning methods, deep generative models, and adaptive learning systems, with application to several fields, from audio to images, from medical to sensor signals. Danilo Comminiello is an elected Member of the IEEE Machine Learning for Signal Processing Technical Committee and of the IEEE Nonlinear Circuits and Systems Technical Committee. He is also the Chair of the IEEE Task Force on Computational Audio Processing. He was an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: REGULAR PAPERS, *Elsevier Digital Signal Processing*, and he is one of the editors of the book *Adaptive Learning Methods for Nonlinear System Modeling* (D. Comminiello and J. C. Principe, eds.), Elsevier 2018. Danilo Comminiello is the General Co-Chair of the 33 rd IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2023), Rome, Italy.

Open Access funding provided by 'Università degli Studi di Roma "La Sapienza"' within the CRUI CARE Agreement