Reparameterization of Lightweight Transformer for On-Device Speech Emotion Recognition

Zixing Zhang, *Senior Member, IEEE*, Zhongren Dong, *Student Member, IEEE*, Weixiang Xu, and Jing Han, *Senior Member, IEEE*

***Abstract*—With the increasing implementation of machine learning models on the edge or Internet of Things (IoT) devices, deploying advanced models on resource-constrained IoT devices remains challenging. Transformer models, a currently dominant neural architecture, have achieved great success in broad domains but their complexity hinders its deployment on IoT devices with limited computation capability and storage size. Although many model compression approaches have been explored, they often suffer from notorious performance degradation. To address this issue, we introduce a new method, namely Transformer reparameterization, to boost the performance of lightweight Transformer models. It consists of two processes: 1) the high-rank factorization (HRF) process in the training stage and 2) the de-HRF (deHRF) process in the inference stage. In the former process, we insert an additional linear layer before the feed-forward network (FFN) of the lightweight Transformer. It is supposed that the inserted HRF layers can enhance the model learning capability. In the later process, the auxiliary HRF layer will be merged together with the following FFN layer into one linear layer and thus recover the original structure of the lightweight model. To examine the effectiveness of the proposed method, we evaluate it on three widely used Transformer variants, i.e., ConvTransformer, Conformer, and SpeechFormer networks, in the application of speech emotion recognition on the IEMOCAP, M^{3}ED, and DAIC-WOZ datasets. Experimental results show that our proposed method consistently improves the performance of lightweight Transformers, even making them comparable to large models. The proposed reparameterization approach enables advanced Transformer models to be deployed on resource-constrained IoT devices.**

***Index Terms*—Artificial Internet of Things (IoT), model compression, speech emotion recognition (SER), transformer reconstruction.**

I. Introduction

With the proliferation of Internet of Things (IoT) devices, deploying artificial intelligence (AI) models on IoT devices becomes imperative to enable real-time intelligent services while preserving privacy [1], [2], [3], [4]. Speech emotion recognition (SER) is a typical application that needs to be deployed on IoT devices, such as smart home assistants and healthcare wearables, due to its privacy and real-time requirements [5], [6], [7], [8]. After the inception of Transformer [9], it has received tremendous success in natural language processing (NLP), speech processing, and computer vision, and continuously achieves state-of-the-art (SOTA) performance in diverse applications, such as emotion recognition, language translation, dialogue systems, and speech recognition [10], [11], [12]. Such great success is not only due to the considerable increase of training data and the optimization of Transformer architecture, but also largely attributed to the remarkable expansion of the model size [10], [13]. For example, the Transformer decoder-based language model of GPT-3 has 175 billion parameters [14] and the model Pangu-Σ has even more than one trillion parameters [15]. The contribution of the model size to the model performance has been comprehensively and empirically investigated in NLP [16].

Meanwhile, in many scenarios, it is needed to deploy models on IoT devices because of the privacy and security, and low-latency requirements [17], [18], [19], [20]. For example, the tasks of emotion recognition and health screening and diagnosis, just to name a few, highly relate to users' personal information, and require keeping the data on their own devices to protect their privacy and security. Besides, other tasks, such as command recognition, face detection, and autonomous driving system are always on and require real-time decision making. However, these IoT devices often lack sufficient storage memory, computational resources, energy, and network bandwidth.

The contradiction between the complexity (e.g., model size and computational operations) of Transformer models and the resource-constrained IoT devices has crucially hindered the on-device deployment of Transformer-style models. To deal with this challenge, plenty of model compression and optimization approaches have been introduced to date [21], [22], including pruning [23], knowledge distillation [24], and matrix decomposition [25]. However, with these model compression approaches, especially when compacting the model into a tiny size one, the performance of the lightweight Transformer often significantly degrades due to the reduction of its learning capability [17], [26].

Received 11 April 2024; revised 25 July 2024; accepted 14 October 2024. Date of publication 18 October 2024; date of current version 6 February 2025. This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515010112; and in part by the Changsha Natural Science Foundation under Grant kq2402082. *(Corresponding author: Jing Han.)*

Zixing Zhang is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China, and also with the Shenzhen Research Institute, Hunan University, Shenzhen 518000, China (e-mail: zixingzhang@hnu.edu.cn).

Zhongren Dong and Weixiang Xu are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: zrdong@hnu.edu.cn; xuweixiang@hnu.edu.cn).

Jing Han is with the Department of Computer Science and Technology, University of Cambridge, CB3 0FD Cambridge, U.K. (e-mail: jh2298@cam.ac.uk).

Digital Object Identifier 10.1109/JIOT.2024.3483232

To address this issue, we propose a novel approach, namely *Transformer reparameterization*, to boost the performance of lightweight Transformer models without any increase of the model size and computational operations. This Transformer reparameterization approach comprises two processes, i.e., the high-rank factorization (HRF) process in the training stage and the de-HRF (deHRF) process in the inference stage. Specifically, in the HRF process, we insert an expanded linear layer before a feedforward network (FFN) of the Transformer. By doing this, it can endow the model with a better learning capability in the training stage due to the increase of the model parameters. This process can be algebraically reversed. Thus, in the inference stage, the inserted linear layer will be removed by merging it together with its followed dense layer into one new dense layer with a mathematical calculation. By this means, the original lightweight Transformer architecture will be reconstructed without adding new parameters and computational operations in the inference stage whilst maintaining the boosted model performance. We evaluate the introduced approach in SER due to its broad applications, such as human–machine interaction, call centers, and mental health tracking [12], [27].

The present work is highly motivated by the "scaling law" of deep models [28], [29], where the model performance has a power-law relationship with the model size. This is not only empirically demonstrated via various tasks and models [28], [29] but also mathematically explained through the theory that an interpolated/over-parameterized classifier is more generalized to the unseen data than a small model [30]. In this work, we focus on the Transformer models and investigate their different modules, including the queries, keys, and values (QKVs) for attention mechanisms, Projection, FFN, classification head (CLS), and their combinations. To the best of our knowledge, this is the first work for enhancing the lightweight Transformers via the lossless model reparameterization strategy.

Our major contributions can be summarized as follows.

1) We, for the first time, introduced a novel reparameterization strategy for enhancing the lightweight Transformers, to the best of our knowledge. It empowers lightweight Transformers with better learning capability in the training process whilst no performance loss in the inference stage, without the price of the model size and computational operation.
2) We comprehensively investigated the proposed reparameterization strategy in different modules of Transformer, including QKV for attention mechanisms, projection, FFN, CLS, and their combinations.
3) We extensively evaluate the effectiveness and generalizability of our approach in the context of SER, a domain with a strong imperative for on-device deployment due to privacy concerns.

The remainder of this article is organized as follows. We first introduce the related work in Section II, and then formalize the problem statement and elaborately describe the proposed method in Section III. After that, the experimental setups and experimental results are given in Section IV. Finally, we draw the conclusion in Section V.

II. Related Work

In this section, we review Transformer-based SER, lightweight Transformers, and structural reparameterization.

A. Transformers-Based Speech Emotion Recognition

For SER, nowadays Transformer-based models have been widely employed due to their capability to capture short- and long-context information. From the feature aspect, it goes from the handcrafted frame-level features (e.g., the frame-level mel-frequency cepstral coefficient [MFCC] or log mel-filter bank energies [LMFB]) [31] and the segment-level features (e.g., IS09 or eGeMAPS) [32], to the raw speech signals that contain the complete information in an end-to-end way [33]. The latter, however, often require a large number of emotional speech data for training.

From the model architecture aspect, early studies often used the classic Transformer for SER [31], [33]. Recently, more and more advanced Transformer structures were designed for SER. For example, an 1-D CNN network was inserted into the classic Transformer to extract information from raw speech signal [33]. Besides, a SpeechFormer model was introduced, which optimized global attention to compute local attention and employed a hierarchical structure to extract frame-, phoneme-, word-, and sentence-level features, resulting in superior performance [34]. Furthermore, a DWFormer model has been proposed, which takes dynamic windows to extract fine-grained temporal features from speech samples for SER [35].

Notably, all these aforementioned studies trained Transformer models from scratch in a supervised learning way. With the advent of pretrained models, nowadays more and more research is focusing on the use of pretrained models for SER [12], [36]. For example, Wang et al. [36] performed partial and overall fine tuning of pretrained models, including Wav2vec 2.0 and HuBERT, to study their feasibility in SER, speaker verification, and spoken language understanding tasks. Wagner et al. [12] conducted a comprehensive evaluation of several pretrained variants of Wav2vec 2.0 and HuBERT for SER. Although these methods show promising performance, their high computational complexity, and particularly their model size, renders them unsuitable for the edge devices.

B. Lightweight Transformers

With the popularity of mobile/edge devices, increasing interest is focusing on the reduction of the computational cost and model size of Transformers and have proposed many compression methods [25], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51]. These methods can be mainly classified into three categories: 1) pruning [37], [38], [39], [40], [41], [42]; 2) knowledge distillation [43], [44], [45], [46]; and 3) matrix decomposition [25], [47], [48], [49], [50], [51].

Pruning removes redundant connections from the model, and thus reduces the model size and computation operations, and speeds up the inference process. The simplified process of pruning refers to Fig. 1(a). Studies on Transformer-based pruning typically concentrate on different modules of

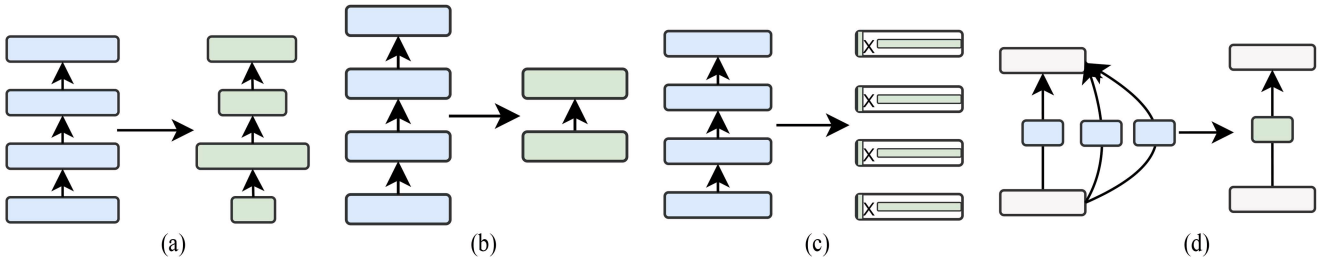

Fig. 1. Four categories of model compression methods. (a) Pruning: Removing some weights or connections from the network. (b) Knowledge Distillation: Transferring knowledge from a large and accurate teacher model to a smaller student model. (c) Matrix Decomposition: Decomposing a large weight matrix into several smaller matrices. (d) Structural Reparameterization: Shrinking a multibranch structure into a single-branch structure, typically employed in convolutional neural networks.

Transformer, such as pruning the attention head [37], [38], pruning the FFN [39], [40], or even directly removing certain layers [41], [42]. Researchers often carefully eliminate a module or layer based on its contribution to the model size, ensuring that the overall performance of the model is not compromised by excessive removal.

Knowledge distillation trains a smaller and less computationally intensive model by transferring knowledge from a large and more accurate model. Its streamlined procedure is illustrated in Fig. 1(b). One of the most basic purposes of Transformer-based distillation work is to integrate the capabilities of the teacher model into a relatively smaller model. DistillBERT [43] and TinyBERT [44] both integrate standard BERT into a model with fewer layers, and MobileBERT [45] integrates standard BERT into a narrower network, with the same number of layers as BERT. Similarly, Distilhubert [46] has the standard structure of HuBERT but fewer layers, retaining almost the same performance as HuBERT. One feature of knowledge distillation is that no matter how small your student model is, we need to train a large teacher model.

Matrix decomposition aims to reduce the number of parameters and computation by decomposing the weight matrix in the model into the product of multiple successive submatrices, and its simplified process refers to Fig. 1(c). Early work on the transformer-based matrix decomposition typically focused on approximating the attention matrix using a low-rank matrix for the self-attention module [25], [47], [48]. Subsequently, researchers went beyond the attention module and performed matrix decomposition on the entire transformer to compress the model, potentially achieving greater compression [49], [50], [51]. Matrix decomposition does not decrease the number of layers in the model but only approximates certain modules in the transformer, such as the self-attention module and the FFN module (FFN_M). Consequently, matrix decomposition methods usually do not compress the model to a significant extent. As the degree of compression increases, the model's performance tends to decrease substantially.

In any scenario, despite the reduction in the size of the network through the aforementioned compression methods, it not only fails to provide the flexibility to design a network with a specific architecture but also does not combine excessive parameterization to enhance the performance of training a compact network [52].

C. Structural Reparameterization

Structural reparameterization [53] is different from all aforementioned methods. It decouples the model training and inference stages, typically employing a multibranch structure during the training process, and effectively reverts back to the original single-branch structure during the inference phase. The simplified process of structural reparameterization refers to Fig. 1(d). Structural reparameterization was previously investigated in CNN networks, such as using asymmetric convolution instead of regular convolutional layers [54], expanding a single regular convolution into multiple convolutional layers [52], and expanding a regular convolution into a multibranch convolution [53]. A common feature of these networks is that the expanded modules can be shrunk back into a single-branch standard convolution in the inference stage, which allows the model to be characterized by both representational power and computational efficiency. To date, no structural reparameterization techniques have been studied for boosting the lightweight Transformer performance.

III. Reparameterization of Lightweight Transformer

In this section, we first formalize the problem statement, then present the framework overview, and finally describe the detailed process and implementation rationale for the Transformer reparameterization.

A. Problem Formulation

We evaluate the proposed Transformer reparameterization in the application of SER. Let $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ be the dataset consisting of n utterances, where x_i denotes the ith utterance and y_i denotes its corresponding emotion label (e.g., neutral, sadness, happiness, fear, anger, surprise, and disgust). The goal is to train a model $\mathcal{M}$ on D to predict the emotion labels of the test utterances.

B. Overview

The overall framework diagram is shown in Fig. 2, where we select three widely used Transformer structures in the speech domain, i.e., ConvTransformer [55], Conformer [56], and SpeechFormer [34]. The introduced Transformer reparameterization includes the HRF process in the training stage and the deHRF process in the inference stage.

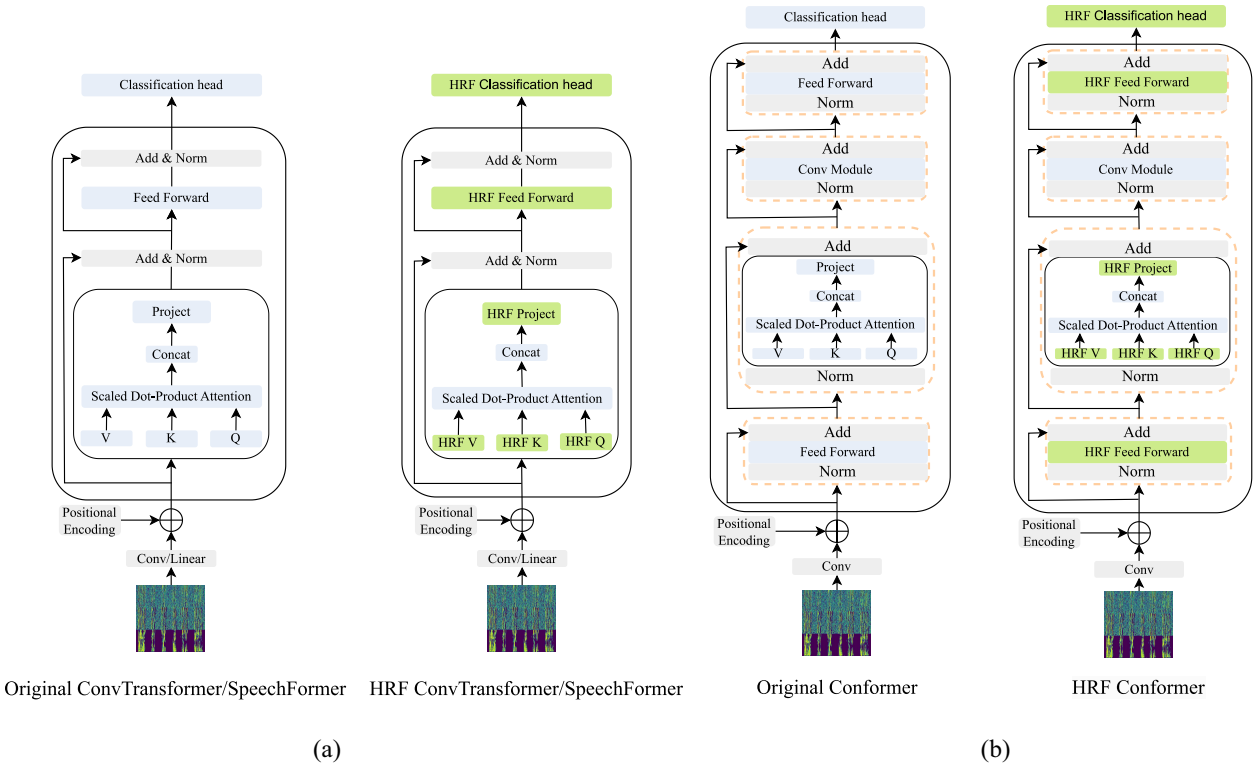

Fig. 2. Framework of Transformer reparameterisation in different modules with the examples of ConvTransformer, SpeechFormer, and Conformer. (a) ConvTransformer/SpeechFormer. (b) Conformer.

Specifically, in the training stage: for ConvTransformer or SpeechFormer, we perform HRF on the QKV module, Project module, FFN_M, CLS module, and all modules (ALL). As the FFN_M is composed of two fully connected (FC) layers, HRF FFN will have three strategies, i.e., FFN-1, FFN-2, and FFN-1 & FFN-2. Detailed information is illustrated in the right subfigure of Fig. 2(a). Similar to ConvTransformer and SpeechFomer, we apply HRF on the same modules for Conformer. Nevertheless, as a Macaron structure is performed for Conformer, i.e., two FFN networks are used before and after the self-attention block, we additionally conduct HRF on the additional FFN_M as well, i.e., FFN_M-1, FFN_M-2, and FFN_M-1 & FFN_M-2. More details about the HRF Conformer can be found in the right subfigure of Fig. 2(b).

In the inference stage: All the HRF modules will be reconstructed and converted back to the original structure, which is presented in the left subfigures of Fig. 2(a) for ConvTransformer or SpeechFormer and Fig. 2(b) for Conformer.

Taking an FFN_M of Transformer for example, Fig. 3 depicts detailed HRF and deHRF processes. In the training phase, three types of HRF expansions are performed, i.e., FC1 only (FFN-1), FC2 only (FFN-2), and both FC1 and FC2 (FFN-1 & FFN-2). In the inference phase, the expanded FC layers are shrunk to a single FC layer by a deHRF process. The principles of HRF and deHRF are explained in Sections III-D and III-E.

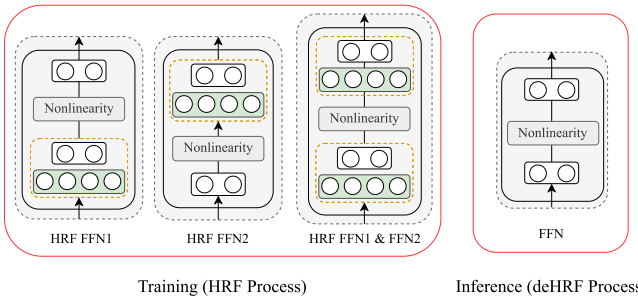

Fig. 3. Detailed illustration of the reparameterization process in the transformer FFN.

C. Linear Over-Parameterization

Before introducing our method, let's first qualitatively discuss the importance of linear overparameterization, which is also the theoretical basis of HRF. Suppose we are learning a linear model parameterized by a matrix W, obtained through training $\min\{L(W)\}$. Now, we use a linear neural network with N layers to implement the learning process, then $W = W_N W_{N-1} \dots W_1$, where W_j is the weight matrix of a specific layer. We use a lower learning rate η for gradient descent, and according to [57], the weight matrix $W = W_N W_{N-1} \dots W_1$ satisfies the dynamics of continuous gradient descent

$$W_T^{j+1} W_{j+1} = W_j W_j^T. \tag{1}$$

Then, the update rule for the end-to-end weight matrix can be expressed in the form of a differential equation

$$W^{t+1} \longleftarrow W^t - \eta \sum_{j=1}^{N} \left[W^t (W^t)^T \right]^{\frac{j-1}{N}} \frac{\mathrm{d}L}{\mathrm{d}W}(W^t) \left[(W^t)^T W^t \right]^{\frac{N-j}{N}} \tag{2}$$

where t is now a discrete time index. $[\cdot]^{([j-1]/N)}$ and $[\cdot]^{([N-j]/N)} j = 1, \ldots, N$ are the fractional power operators defined on the positive semi-definite matrices. We left multiply by a matrix $[W^t(W^t)^T]$, right multiply by a matrix $[(W^t)^T W^t]$, and then sum over j, which is essentially a special form of preconditioning that promotes movement along the optimization direction and can be used for optimizing deep networks. More importantly, the above update rule does not depend on the width of the hidden layers in the linear neural network, but on the depth (N). This can be explained as overparameterization promoting movement along the direction already taken for optimization and can thus be viewed as a form of acceleration [57].

In addition to the qualitative discussion of overparameterization, in practice, the performance of large language models strongly depends on the number of model parameters N (excluding embeddings), the size of the dataset D, and the computational resources C used for training [29]. As long as we simultaneously increase N and D, the performance will predictably improve, which is known as the scale law of large language models [28], [58]. The scale law in practice demonstrates the importance of overparameterization for the model performance.

D. High-Rank Factorization

Theoretically, with the HRF processing, an FC layer that serves as the foundational layer in Transformer can be expanded into multiple FC layers. Assuming an FC layer with m input neurons and n output neurons, the output Y can be represented as

$$Y = \sigma(WX + b) \tag{3}$$

where X is the input, Y is the output of the FC layer, $\sigma()$ is the activation function, $W \in R^{m \times n}$ is the weight matrix, and $b \in R^n$ is the bias. The HRF process is defined as follows:

$$W = W_N \times W_{N-1} \times \ldots W_1. \tag{4}$$

N can be arbitrarily large. But, due to the computational cost of FC layers, we set the maximum N to 3 in this article. When $N = 3$, $W_3 \in R^{m \times r}$, $W_2 \in R^{r \times r}$, $W_1 \in R^{r \times n}$, and r is the expansion scale and can be set to two, four, or eight times the second dimension of W, i.e., $r = 2n$, 4n, or 8n. Notably and importantly, when expanding one FC layer into multiple FC layers, no activation functions are applied after each extended HRF layer.

This process is similar to the low-rank matrix factorization/de-composition (LRF) that is widely used for the model compression [17]: given any matrix $W^* \in R^{m \times n}$ of full rank $r^* \leq \min\{m, n\}$, it can be decomposed into $W^* \simeq AH$, where $A \in R^{m \times r^*}$ and $H \in R^{r^* \times n}$. When r^* is much smaller than m or n, the space complexity can be significantly reduced from $O(mn)$ to $O(r^*(m + n))$. This is because both LRF and HRF use multiple matrices to represent a single matrix with the difference that LRF has $r^* \leq \min\{m, n\}$ whilst HRF normally has $r > \max\{m, n\}$, we refer to our proposed method as HRF. The details of HRF are shown in Algorithm 1.

Algorithm 1: Pseudocode of the HRF

Input:
m: Input dimension of FC layer
n: Output dimension of FC layer
r: Expansion ratio of HRF layer
Output:
HRF: High Rank Factorization layer

procedure HRF(m, n, r)
  $FC \leftarrow Linear(m, n, bias = True)$
  $HRF1 \leftarrow Linear(m, r \times n, bias = True)$
  $HRF2 \leftarrow Linear(r \times n, n, bias = True)$
  $HRF \leftarrow Sequential(HRF1, HRF2)$
  return *HRF*
end procedure

Algorithm 2: Pseudocode of the deHRF

Input:
FC: Original FC layer
HRF: High Rank Factorization layer
Output:
FC: Reconstructed FC layer

procedure DEHRF(FC, HRF)
  $HRF1, HRF2 \leftarrow HRF$
  $W1, b1 \leftarrow HRF1.weight, HRF1.bias$
  $W2, b2 \leftarrow HRF2.weight, HRF2.bias$
  $W \leftarrow MatrixMultiplication(W2, W1)$
  $b \leftarrow MatrixMultiplication(W2, b1) + b2$
  $FC.weight.data \leftarrow W$
  $FC.bias.data \leftarrow b$
  return *FC*
end procedure

E. De-High-Rank Factorization

In the training phase, we employ HRF in the model; while in the inference phase, we reconstruct the HRF model back to its original size and structure. The reconstruction process involves a chain of matrix calculation as no nonlinear activation functions are conducted between the adjacent FC layers. Given one FC layer with one HRF layer expansion (i.e., $N = 1$) for example

$$Y = \sigma(WX + b) \tag{5}$$

where X is the input and Y is the output. After applying HRF to the FC layer, we obtain Y^* that is calculated using the following equation:

$$\begin{aligned} Y^* &= \sigma(W_2(W_1X + b_1) + b_2) \\ &= \sigma((W_2W_1)X + (W_2b_1 + b_2)). \end{aligned} \tag{6}$$

Let $W = W_2W_1$, $b = W_2b_1 + b_2$, then $Y = Y^*$. By doing this calculation, we can contract the two layers back to one original FC layer. Algorithm 2 describes the details of the deHRF process. To be noted, when $N > 1$, a similar chain of matrix calculation can be conducted to reconstruct the HRF modules.

IV. Experiments and Results

In this section, we introduce the selected datasets, experiment implementation details, and selected Transformer models, and present the results followed by extensive discussions and ablation studies.

A. Datasets

We evaluated the introduced Transformer reparameterization on two multimodal emotional datasets—IEMOCAP [59], M^3ED [60], and DAIC-WOZ [61]. In this work, we mainly focus on the speech signals for emotion recognition.

IEMOCAP is the most widely exploited dataset for SER. It comprises 12.46 h of audio data, divided into five sessions with one male and one female speaker each. The conversations were segmented into utterances and annotated by at least three annotators using the following discrete categories: anger, happiness, sadness, neutrality, excitement, frustration, fear, surprise, and others. For this study, we focused on the first four categories, and merged "excitement" into "happiness." Following previous research [62], [63], we used the first four sessions for training and the final session for testing. To reduce randomness, we utilized five random seeds for training and testing, and averaged the final results over the five tests. We took unweighted accuracy (UA), weighted accuracy (WA), and weighted average F1 (WF1) as performance metrics for the model evaluation.

M^3ED is a recently released and the first Chinese multimodal sentiment dialogue dataset, comprising 990 dyadic emotional dialogues from 56 different TV series, with a total of 9082 turns and 24 449 utterances. The dataset was annotated with seven emotional categories (i.e., happy, surprised, sad, disgusted, angry, fearful, and neutral) at the verbal level, including sound, visual, and textual patterns. We followed the default data partition strategy and used five random seeds for training, validation, and testing, with the final results averaged over the five tests. We took WF1 as the performance metric to evaluate models. Note that, the metrics of UA and WA were not considered for M^3ED mainly due to the alignment with other SOTA approaches for performance comparison. The detailed emotion distribution for the IEMOCAP and M3ED datasets are listed in Table I.

The DAIC-WOZ dataset, derived from The AVEC2017 series depression detection task, comprised 189 segments of clinical interviews. The dataset was divided into 107 segments for training, 35 segments for validation, and 47 segments for testing. These interviews were conducted to aid in diagnosing conditions, such as depression.

Each segment in the dataset included audio and video features, alongside transcripts of the interviews. Segment durations ranged from 7 to 33 min, with an average duration of 16 min. To ensure fair comparisons with other SOTA works, we utilized the training and validation sets and reported macro average F1 (MF1) scores on the validation set.

For training convenience, we extracted individual patient voices from the provided transcripts and segmented them into 10 s clips. The training set consisted of 5084 clips, while the validation set contained 1931 clips. Detailed distribution of the dataset is presented in Table II.

TABLE I
Emotion Distribution of IEMOCAP and M^3ED

Emotion	IEMOCAP			M^3ED		
	Train	Val	Test	Train	Val	Test
neutral	1,324	-	384	7,130	1,043	1,855
happy	1,194	-	442	1,626	303	358
sad	839	-	245	2,734	489	734
anger	933	-	170	3,816	682	736
surprise	-	-	-	696	120	235
disgust	-	-	-	1,145	134	218
fear	-	-	-	280	50	65
Total	4,290	-	1,241	17,427	2,821	4,201

TABLE II
Data Distribution of DAIC-WOZ

Class	Train	Val	Test
N-Dep.*	87	28	38
Dep.*	21	7	9
Total Number of participants	107	35	47
Total Number of clips	5084	1931	-

[1] *All participants were assigned into one of two classes, depressed (Dep.) or non-depressed (N-Dep.), based on the PHQ-8 scores.

B. Implementation Details

Generally, there are three types of models: 1) the original model (the original large one); 2) the lightweight model (a lighter version of the original one); and 3) the HRF model (the lightweight model with an HRF process). We implemented all models under the PyTorch framework.

The initial learning rate for the original and lightweight ConvTransformer and Conformer were the same with 1e-3; while for the original and lightweight Speechformer they were 1e-3 and 5e-4, respectively. The learning rate decays to half when the loss does not decrease. In addition, the initial learning rate for the HRF model was identical to the corresponding lightweight model. All models used the AdamW optimizer, with the decay rate set to 1e-6 and the epoch set to 120 (for IEMOCAP) or 100 (for M^3ED and DAIC-WOZ). The original ConvTransformer, Conformer, and SpeechFormer had the structures of [8, 80, 320], [4, 80, 320], and [8, 80, 64], where the three values in each bracket denote the number of Transformer blocks (n_{layer}), the dimensionality of the input data for each Transformer block (d_{model}), and the dimensionality of the first FC layer in FFN (d_{ffn}), respectively. These original models were then compressed into tiny ones for the on-device deployment purpose, with new structures of [1, 16, 4], [1, 16, 2], and [1, 16, 4] for lightweight ConvTransformer, Conformer, and SpeechFormer, respectively. All models hold four attention heads. Notably, all HRF models had the same parameters as their corresponding lightweight models during the inference stage. The default activation functions for ConvTransformer and SpeechFormer are ReLU, and for Conformer it is Swish.

TABLE III
PERFORMANCE OF LIGHTWEIGHT *ConvTransformer* WHEN HRF IS APPLIED TO ITS DIFFERENT MODULES ON THE IEMOCAP, M^{3}ED, AND DAIC-WOZ DATASETS

which modules use HRF [×8]	IEMOCAP					M^{3}ED			DAIC-WOZ		
	WA	UA	WF1	Params	FLOPs	WF1	Params	FLOPs	MF1	Params	FLOPs
Original	.589	.602	.586	801K	226M	.399	595K	160M	.572	801K	454M
Lightweight	.552	.567	.550	9K	7M	.372	9K	6M	.530	9K	14M
FFN1	.562	.578	.560	9K	7M	.386	9K	6M	.573	9K	14M
FFN2	**.580**	**.589**	**.580**	9K	7M	.382	9K	6M	.558	9K	14M
FFN1 & FFN2	.566	.582	.565	9K	7M	.381	9K	6M	**.580**	9K	14M
QKV	.566	.583	.562	9K	7M	.380	9K	6M	.551	9K	14M
Project	.566	.583	.565	9K	7M	.382	9K	6M	**.580**	9K	14M
CLS	.572	.582	.570	9K	7M	**.387**	9K	6M	.551	9K	14M
ALL	.565	.571	.563	9K	7M	.383	9K	6M	.530	9K	14M

TABLE IV
PERFORMANCE OF LIGHTWEIGHT *Conformer* WHEN HRF IS APPLIED TO ITS DIFFERENT MODULES ON THE IEMOCAP, M^{3}ED, AND DAIC-WOZ DATASETS

which modules use HRF [×8]	IEMOCAP					M^{3}ED			DAIC-WOZ		
	WA	UA	WF1	Params	FLOPs	WF1	Params	FLOPs	MF1	Params	FLOPs
Original	.572	.572	.571	809K	227M	.378	323K	50M	.633	814K	458M
Lightweight	.524	.558	.520	10K	7M	.368	11K	6M	.533	10K	15M
FFN1	.545	.556	.543	10K	7M	.376	11K	6M	.595	10K	15M
FFN2	.545	.560	.542	10K	7M	.375	11K	6M	.571	10K	15M
FFN1 & FFN2	.545	.555	.542	10K	7M	.371	11K	6M	.592	10K	15M
FFN_M1	**.554**	.557	**.553**	10K	7M	**.379**	11K	6M	.603	10K	15M
FFN_M2	.545	.553	.542	10K	7M	.376	11K	6M	.596	10K	15M
FFN_M1 & FFN_M2	.542	**.561**	.540	10K	7M	.372	11K	6M	.583	10K	15M
FFN1 & FFN2 & FFN_M1 & FFN_M2	.551	.555	.550	10K	7M	.375	11K	6M	**.644**	10K	15M
QKV	.536	.535	.532	10K	7M	.370	11K	6M	.608	10K	15M
Project	.547	.560	.545	10K	7M	.375	11K	6M	.603	10K	15M
CLS	.535	.555	.532	10K	7M	.367	11K	6M	.540	10K	15M
ALL	.548	.558	.547	10K	7M	.367	11K	6M	.570	10K	15M

Regarding to the acoustic features, we followed previous work [64] and extracted 78-D features from speech signals using 26 filter banks with a window length of 25 ms and a hop size of 10 ms. These 78-D features comprise 26 original LMFBs, and their first and second derivatives. Please note that we do not directly employ the pretrained models for feature extraction, as these pretrained models are too large to be deployed on the device, or it is too sensitive to send the users' private data to the cloud. Besides, we segmented the sentences into 5 and 4 s for IEMOCAP and M^{3}ED datasets, respectively, and pursued a zero-filling strategy if the sentence was insufficiently long.

C. Baseline Models

Three typical Transformer models in the speech domain were selected to examine the performance of the introduced Transformer reparameterization and are listed as follows.

1) *ConvTransformer [55]:* A classic Transformer architecture for speech recognition. On the top of the classic Transformer, it employs a VGG-style convolutional network that is able to potentially extract high-level representations of speech signals and their relative positions.
2) *Conformer [56]:* A frequently used model in speech domain. The Conformer is a Macaron-style structure which exploits FFN before and after the self-attention module.
3) *SpeechFormer [34]:* A latest SOTA Transformer variant that extracts different levels (i.e., frame, phoneme, and word) of acoustic representations for SER. Compared to the original SpeechFormer, we made slight modifications. That is, we insert a linear layer to map the high-dimensional features of the model inputs into a smaller one, which makes the d_{model} reduced.

D. Results and Discussions

As aforementioned in Section III-B, we investigated the modules of FFN, QKV, Project, CLS, and ALL for both ConvTransformer and SpeechFormer, and the one additional module of FFN-M for Conformer. Again, since an FFN is composed of two FC layers, it can be divided into FFN1, FFN2, and FFN1 & FFN2. It is the same with FFN-M (i.e., FFN-M1, FFN-M2, and FFN-M1 & FFN-M2).

Tables III–V present the results evaluated on the IEMOCAP, M^{3}ED and DAIC-WOZ datasets for ConvTransformer, Conformer, and SpeechFormer, respectively. All these results were obtained when the HRF expansion ratio is eight. First, we can see that the original Transformer models, some of which have already been compressed, have hundreds of thousands of parameters and FLOPs (i.e., computational complexity). With such kinds of model parameters and FLOPs, it is challenging to deploy these models on IoT devices.

TABLE V
PERFORMANCE OF LIGHTWEIGHT *SpeechFormer* WHEN HRF IS APPLIED TO ITS DIFFERENT MODULES ON THE IEMOCAP, M^{3}ED, AND DAIC-WOZ DATASETS

which modules use HRF [×8]	IEMOCAP					M^{3}ED			DAIC-WOZ		
	WA	UA	WF1	Params	FLOPs	WF1	Params	FLOPs	MF1	Params	FLOPs
Original	.599	.605	.598	323K	50M	.383	874K	135M	.720	327K	100M
Lightweight	.532	.533	.528	3K	1M	.358	3K	1M	.620	3K	3M
FFN1	.537	.538	.533	3K	1M	**.363**	3K	1M	.643	3K	3M
FFN2	**.545**	**.547**	**.543**	3K	1M	.360	3K	1M	.669	3K	3M
FFN1 & FFN2	.506	.491	.503	3K	1M	.354	3K	1M	**.735**	3K	3M
QKV	.534	.532	.531	3K	1M	.361	3K	1M	.709	3K	3M
Project	.539	.529	.536	3K	1M	.362	3K	1M	.720	3K	3M
CLS	.530	.525	.527	3K	1M	.361	3K	1M	.709	3K	3M
ALL	.536	.535	.531	3K	1M	.362	3K	1M	.694	3K	3M

For this reason, we largely compressed the original ConvTransformer, Conformer, and SpeechFormer into tiny ones by reducing the depth and width of these models (shown in the second row of Tables III–V) to make it more feasible under the on-device scenario. However, we also observe that their performance is greatly degraded, which is consistent with our expectations.

When applying HRF to the three selected Transformers, it is found that HRF-plugged models outperform the ones without HRF layers in most cases. Even in some cases, they are comparable to the original models. Taking ConvTransformer for example, the best WF1s of the HRF model on IEMOCAP and M^{3}ED datasets are 0.580 and 0.387, which are obviously higher than 0.550 and 0.372 for the lightweight model and are close to 0.586 and 0.399 for the original model, respectively. Similarly, on the DAIC-WOZ dataset, the best MF1 score of the HRF model is 0.580, markedly higher than 0.533 for the lightweight model, and exceeds 0.572 for the original model. Therefore, the Transformer reparameterization process can generally well fill the performance gap between the large models and their lightweight versions. This is largely attributed to the expansion of the model size which plays an important role in the model performance as discussed in the previous work [65], [66].

In addition, we also notice that SpeechFormer performs less well compared to the other two Transformers on IEMOCAP and M^{3}ED datasets. This is possibly due to the specific design of SpeechFormer with four stages (each stage has a different number of layers of Transformer blocks) to extract features at different levels of speech (i.e., frames, phonemes, words, and utterances). However, the lightweight model has only one stage and can only extract features at the frame level, which leads to a decrease in the overall learning ability. On the contrary, SpeechFormer performs the best on the DAIC-WOZ dataset. This is because the DAIC-WOZ dataset has too few samples, and the parameter count of ConvTransformer and Conformer is approximately three times that of SpeechFormer, making it prone to overfitting issues. SpeechFormer, on the other hand, is easier to train.

Moreover, when comparing different modules of FFN, QKV, Project, CLS, and ALL with HRF, we can generally find that the HRF FNN performs better than the other modules in five out of six cases with ConvTransformer, Conformer, and SpeechFormer on IEMOCAP, M^{3}ED and DAIC-WOZ datasets. This finding basically suggests that FFN layers play a vital role in Transformer, which consecutively maps the representations from the low-level space to a high-level space in a nonlinear way. However, it is worth noting that FFN often holds a large ratio of the total parameters of Transformers due to its full-connection characteristics. Therefore, it is important to balance the size of FFN and its overall performance, which further raises the importance of the introduced reparameterization approach.

We further observe that the HRF FFN-2 is superior to the HRF FFN-1 or HRF FFN-1 & FFN-2 in most cases. This indicates that mapping representations from high dimension to low dimension often lose a lot of information. This lost information can be well captured by inserting an HRF layer. Besides, it can be seen that when inserting an HRF layer for QKV attention modules, less performance improvement is observed. This is possible because of the redundancy of classic QKV matrics, which is consistent with previous findings in this work [37], [38]. Furthermore, we also notice that combining different modules for HRF does not seem to help much. One potential assumption might be that the inserted HRF layers have no activation functions, and thus they have limited capability for the model to learn nonlinearly. Such a kind of linear transformation might not need several times in one Transformer block.

In addition to the above experiments, we also investigated how well the selected Transformer models are to confirm that these models are reliable representatives for SER. We compared the original three models with some other recent studies on SER. It is important to note that our proposed models are designed for mobile/edge device scenarios, which prevents us to exploit large pretrained models, such as Wav2vec and HuBERT, to extract high-level representations. To ensure fair comparisons between these models, we only considered handcrafted acoustic features, including Mel spectrum, log-mel spectrum, LMFB, MFCC, etc. The results of these comparisons are shown in Tables VI–VIII for the IEMOCAP, M^{3}ED, and DAIC-WOZ datasets. As seen in Tables VI and VIII, when using manual features, the original models can achieve comparable performance to other work for IEMOCAP and DAIC-WOZ. However, for M^{3}ED, our baseline was not as good as the SOTA model [60]. This is due to the pretrained

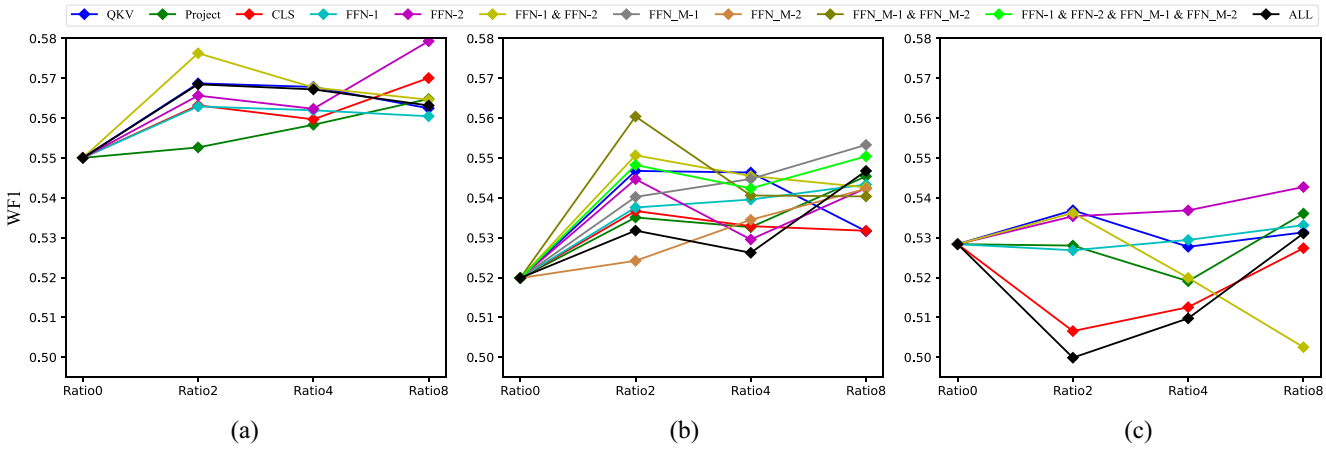

Fig. 4. Results of WF1 when applying different *expansion ratios* of HRF to different modules of ConvTransformer (a), Conformer (b), or SpeechFromer (c) on the *IEMOCAP* dataset.

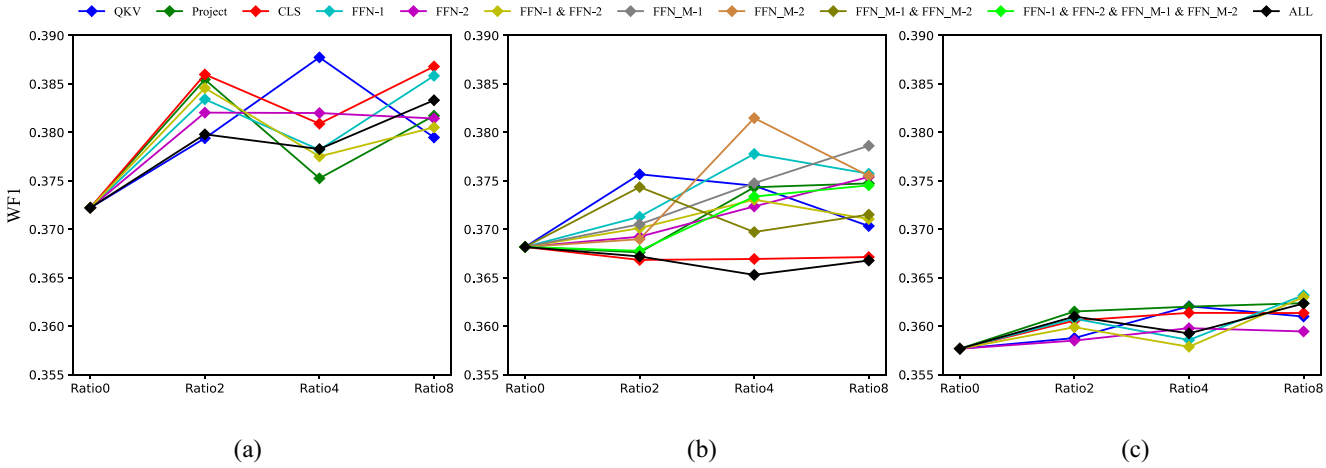

Fig. 5. Results of WF1 when applying different *expansion ratios* of HRF to different modules of ConvTransformer (a), Conformer (b), or SpeechFromer (c) on the *M^3ED* dataset.

TABLE VI
PERFORMANCE COMPARISON BETWEEN OUR EXEMPLIFIED TRANSFORMERS AND OTHER SOTA MODELS ON THE *IEMOCAP* DATASET WHEN USING CLASSIC ACOUSTIC FEATURES

Methods	Features	WA	UA	WF1
SpeechFormer [34]	Log-mel	.582	.598	-
CNN-ELM-attention [67]	Spectrogram	.613	.604	-
SAMS [64]	LMFB	.615	.596	-
BLSTM+attention [62]	IS10	.560	.564	-
our ConvTansformer	LMFB	.589	.602	.585
our Conformer	LMFB	.573	.572	.571
our SpeechFormer	LMFB	.599	.605	.598

TABLE VII
PERFORMANCE COMPARISON BETWEEN OUR EXEMPLIFIED TRANSFORMERS AND OTHER SOTA MODELS ON THE *M^3ED* DATASET WHEN USING HIGH-LEVEL REPRESENTATIONS EXTRACTED FROM PRETRAINED MODELS OR CLASSIC ACOUSTIC FEATURES

Methods	Features	WF1
M^3ED [60]	Wav2vec 2.0	.480
our ConvTransformer	HuBERT	.506
our Conformer	HuBERT	.491
our SpeechFormer	HuBERT	.505
our ConvTransformer	LMFB	.383
our Conformer	LMFB	.378
our SpeechFormer	LMFB	.399

model utilized for feature extraction in [60]. For a fair performance comparison, we further extracted speech features using HuBERT, and the results of all three selected baseline models were better than the models in [60]'s. All these results indicate the reliability of the selected baselines.

E. Ablation Studies

Impact of Different Expansion Ratios (r) of HRF: In order to examine the impact of HRF expansion at various ratios on the results, we conducted expansions of 0 (ratio 0 and original model), 2 (ratio 2), 4 (ratio 4), and 8 (ratio 8) for all lightweight models in different modules. The obtained WF1 versus different expansion ratios of HRF for the three Transformers with different modules on the IEMOCAP, M^{3}ED and DAIC-WOZ datasets are plotted in Figs. 4–6. When varying the expansion ratio, we can see that the HRF Transformers outperform their lightweight versions in most cases. This finding shows the robustness of the Transformer

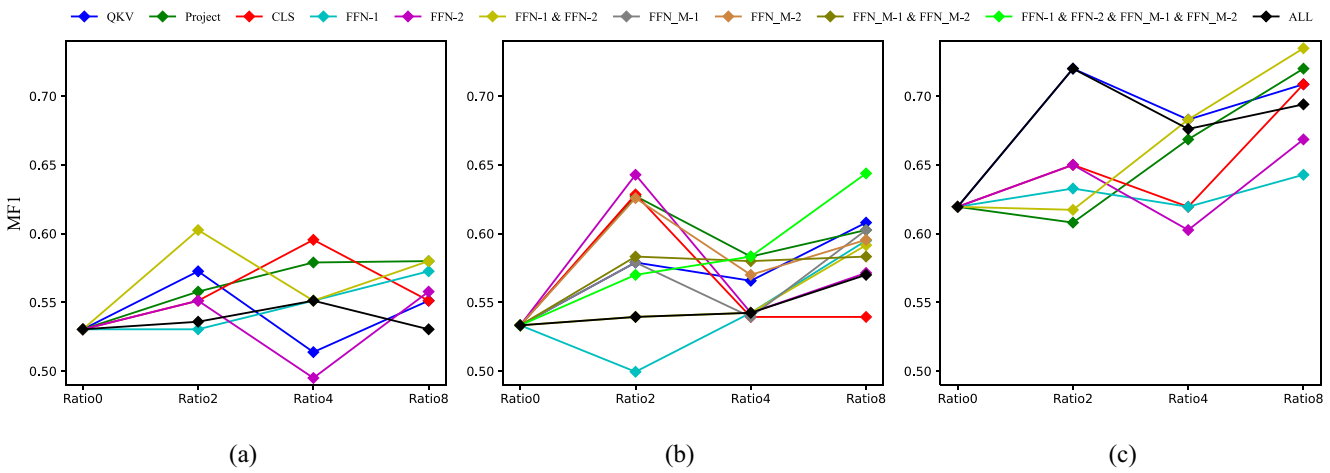

Fig. 6. Results of MF1 when applying different *expansion ratios* of HRF to different modules of ConvTransformer (a), Conformer (b), or SpeechFromer (c) on the *DAIC-WOZ* dataset.

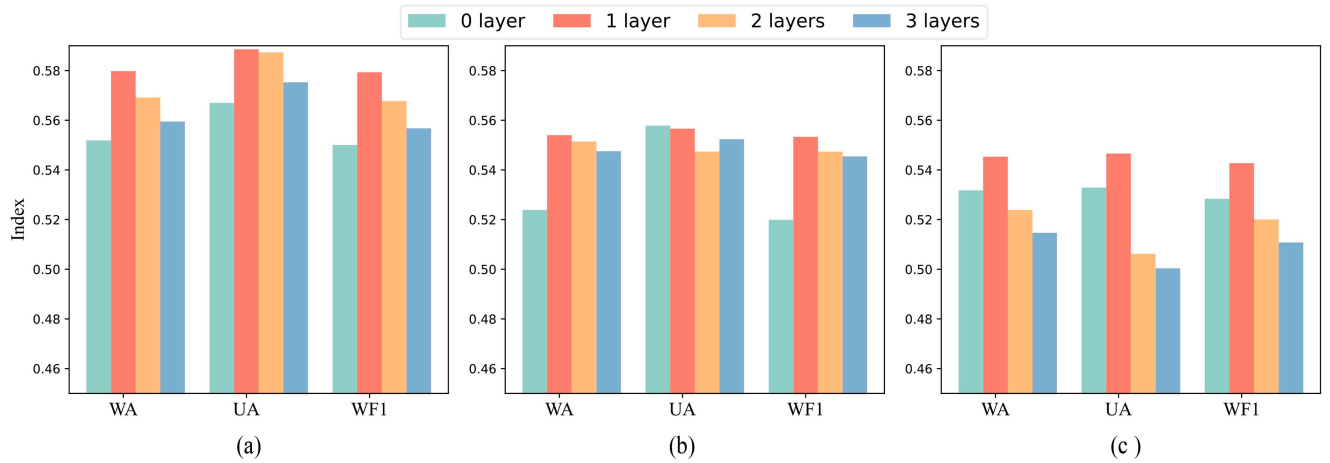

Fig. 7. Results when applying different *numbers of HRF layer* to the second feedforward layers of ConvTransformer (a), Conformer (b), or SpeechFromer (c) on the *IEMOCAP* dataset.

TABLE VIII
PERFORMANCE COMPARISON BETWEEN OUR EXEMPLIFIED TRANSFORMERS AND OTHER SOTA MODELS ON THE *DAIC-WOZ* DATASET WHEN USING CLASSIC ACOUSTIC FEATURES

Methods	Features	MF1
FVTC-CNN [68]	FVTC	.640
EmoAudioNet [69]	MFCC+Spectrogram	.653
Hybrid CNN-SVM [70]	Spectrogram	.680
CNN-LSTM [71]	COVAREP	.610
SpeechFormer [34]	Log-mel	.627
our ConvTransformer	LMFB	.572
our Conformer	LMFB	.633
our SpeechFormer	LMFB	.720

reparameterization approach. Additionally, the different ratios display a slight double-increase trend, i.e., when $r = 2$, 4, or 8, the performance shows an increase, decrease, and increase. We hypothesize that this outcome indicates the phenomenon of double descent [72], [73], which has been proven to exist in image classification tasks. That is, the test error decreases, increases, and finally decreases again as the number of model parameters increases.

Impact of Different Numbers of HRF Layers (n)*:* To quantitatively analyze the impact of different numbers of HRF layers, we extend one HRF layer into two and three layers. To reduce the experiments, we took the best experimental setting for ConvTransformer, Conformer, and SpeechFormer. That is, we fixed the HRF expansion ratio to eight in all cases. Then, for IEMOCAP, we applied multiple HRF layers to the modules of FFN-2, FFN_M1, and FFN-2, for ConvTransformer, Conformer, and SpeechFormer, respectively; whilst for M^{3}ED and DAIC-WOZ, we applied HRF to FFN-1, FFN_M1, and FFN-1 for ConvTransformer, Conformer, and SpeechFormer, respectively. The results versus different numbers of HRF layers are illustrated in Figs. 7–9. The results indicate that extending only one layer achieves the best results. This means that without a nonlinear activation function, one HRF layer can be sufficiently well-trained. Increasing the depth of HRF layers may bring more noise and confusion into the system when doing gradient backpropagation.

Impact of Different Activation Functions for HRF: Different activation functions were used for various Transformer variants. We tested four commonly used activation functions

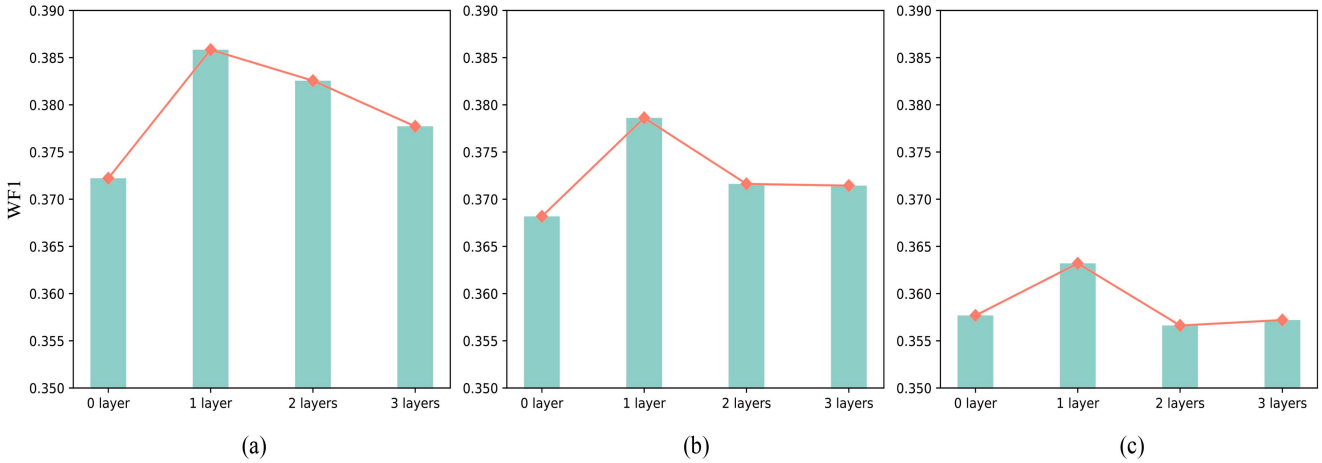

Fig. 8. Results when applying different *numbers of HRF layer* to the second feedforward layers of ConvTransformer (a), Conformer (b), or SpeechFromer (c) on the M^3ED dataset.

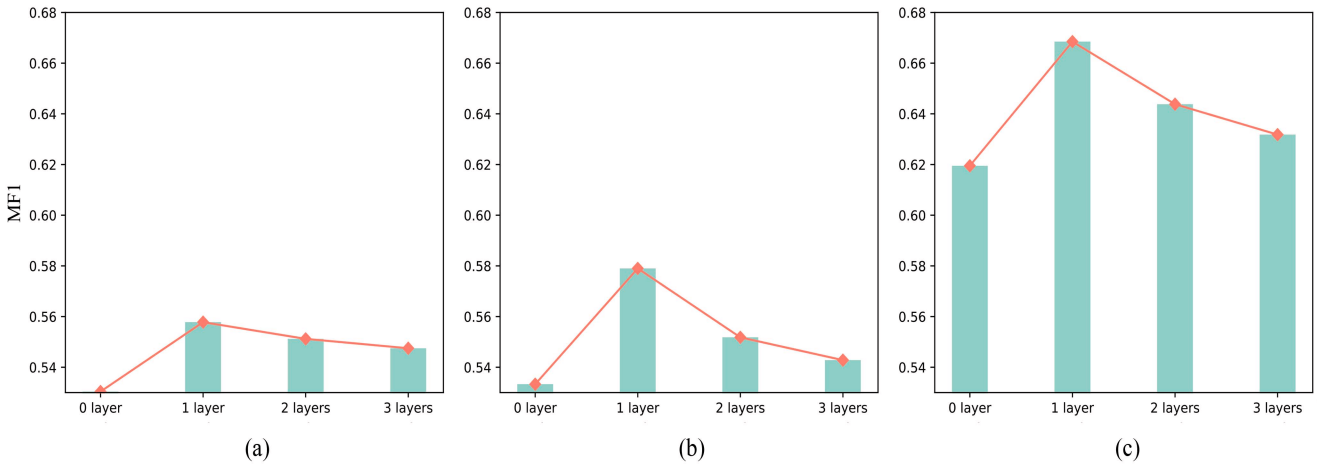

Fig. 9. Results when applying different *numbers of HRF layer* to the second feedforward layers of ConvTransformer (a), Conformer (b), or SpeechFromer (c) on the *DAIC-WOZ* dataset.

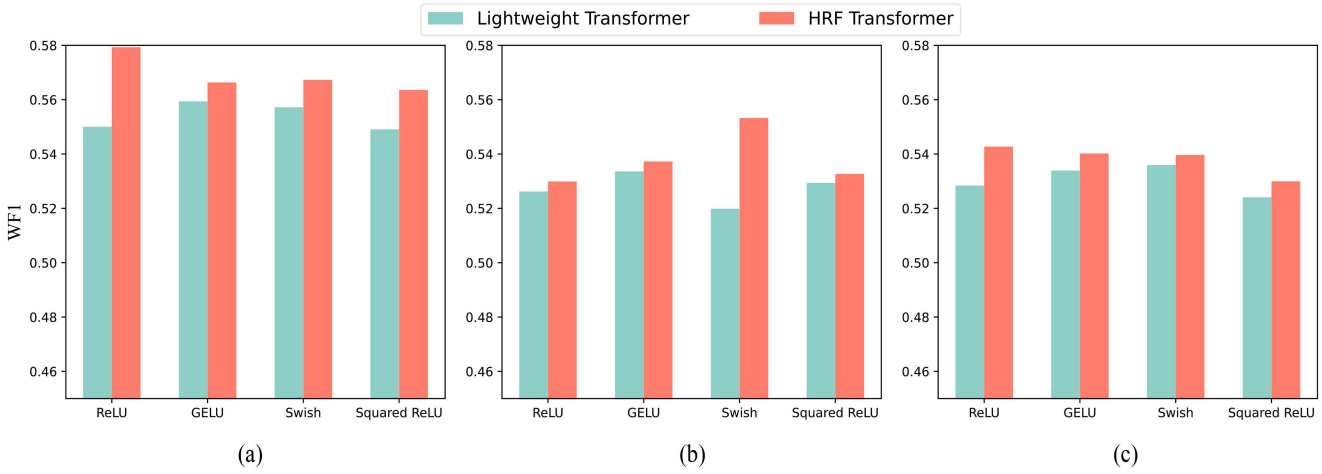

Fig. 10. Performance of reparameterized lightweight ConvTransformer (a), Conformer (b), or SpeechFromer (c) with diverse *activation functions* on the *IEMOCAP* dataset.

for the FFN layer, i.e., ReLU, GELU, Swish, and Squared ReLU. The performance comparison between the lightweight Transformer and the HRF Transformer under different activation functions is depicted in Figs. 10–12. It was observed that under all scenarios the HRF Transformer outperform the lightweight Transformer. This finding further demonstrates the robustness of the Transformer reparameterization approach.

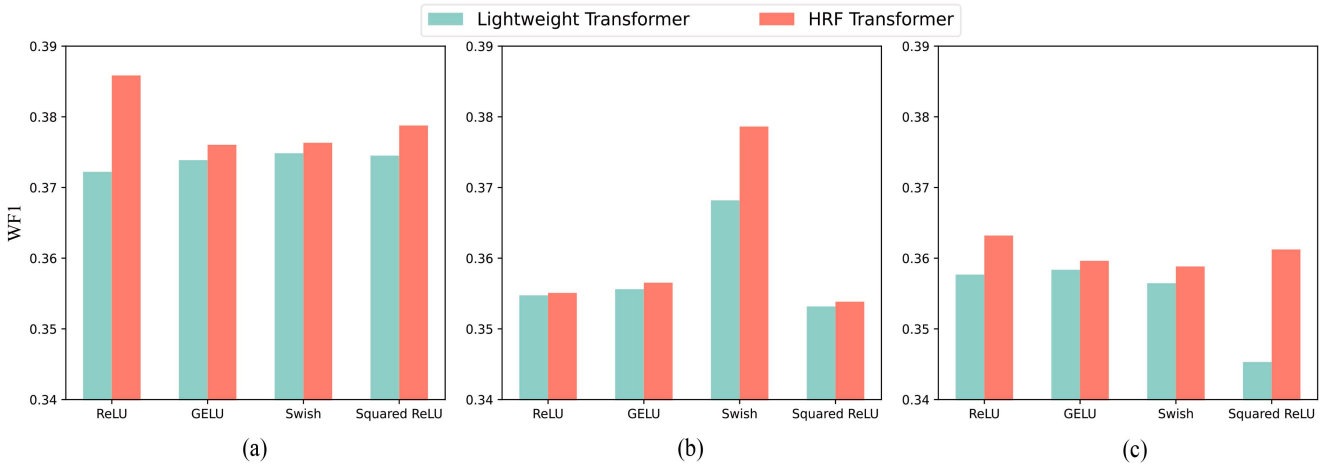

Fig. 11. Performance of reparameterized lightweight ConvTransformer (a), Conformer (b), or SpeechFromer (c) with diverse *activation functions* on the *M*^{3}*ED* dataset.

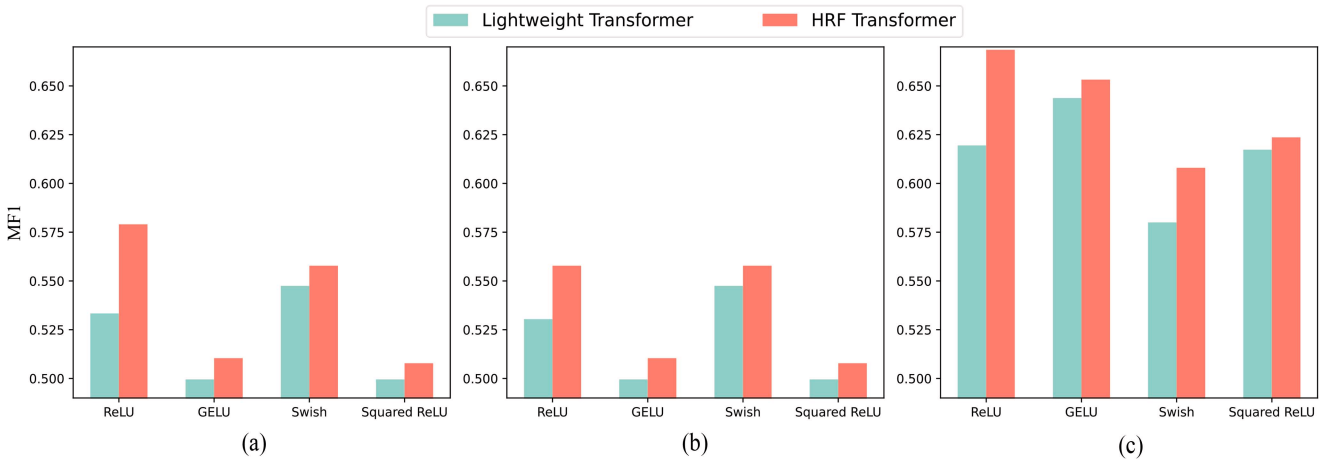

Fig. 12. Performance of reparameterized lightweight ConvTransformer (a), Conformer (b), or SpeechFromer (c) with diverse *activation functions* on the *DAIC-WOZ* dataset.

V. Conclusion

In the present article, we introduce a structure reparameterization method to boost the performance of compressed Transformers. The Transformer architecture can be expanded via an HRF process into a bigger model in the training, which helps increase the model learning capability. In the inference, the expanded model is shrunk into the original one via a reverse mathematical calculation, while retaining the boosted model performance. Our approach is validated on the IEMOCAP, M^{3}ED, and DAIC-WOZ datasets for the SER task. Experimental results show the effectiveness of the structure reparameterization method, which can enhance the lightweight Transformers, and even can make them comparable to larger models with significantly more parameters. The results also empirically reveal the robustness via investigating different numbers of expansion ratios and layers, and different activation functions.

Although the initial findings are promising, further investigation of the introduced method is necessary to unlock its full potential. First, except for the speech modality in this work, other modalities of text or video will be explored as well for multimodal learning in future work. Second, we will extend the evaluation task into a broader range, such as image recognition, speech recognition, and target detection.

References

[1] V. Sharma, T. G. Tan, S. Singh, and P. K. Sharma, "Optimal and privacy-aware resource management in artificial intelligence of things using osmotic computing," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3377–3386, May 2022.

[2] J. Zhang and D. Tao, "Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7789–7817, May 2021.

[3] H. Cai et al., "Enable deep learning on mobile devices: Methods, systems, and applications," *ACM Trans. Design Autom. Electron. Syst.*, vol. 27, no. 3, pp. 1–20, Mar. 2022.

[4] Y. Sun, T. Chen, Q. V. H. Nguyen, and H. Yin, "TinyAD: Memory-efficient anomaly detection for time-series data in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 824–834, Jan. 2024.

[5] R. Chatterjee, S. Mazumdar, R. S. Sherratt, R. Halder, T. Maitra, and D. Giri, "Real-time speech emotion analysis for smart home assistants," *IEEE Trans. Consum. Electron.*, vol. 67, no. 1, pp. 68–76, Feb. 2021.

[6] Z. Tariq, S. K. Shah, and Y. Lee, "Speech emotion detection using IoT based deep learning for health care," in *Proc. 6th Conf. Big Data (Big Data)*, 2019, pp. 4191–4196.

[7] D. D. Olatinwo, A. M. Abu-Mahfouz, G. P. Hancke, and H. C. Myburgh, "IoT-enabled WBAN and machine learning for speech emotion recognition in patients," *Sensors*, vol. 23, no. 6, p. 2948, Mar. 2023.

[8] M. S. Hossain and G. Muhammad, "Emotion-aware connected healthcare big data towards 5G," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2399–2406, Aug. 2018.

[9] A. Vaswani et al., "Attention is all you need," in *Proc. 31th Annu. Conf. Neural Inf. Process. Syst. (NeuIPS)*, 2017, pp. 5998–6008.

[10] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Sci. China Technol. Sci.*, vol. 63, no. 10, pp. 1872–1897, Sep. 2020.

[11] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, no. 10s, pp. 1–41, Jan. 2022.

[12] J. Wagner et al., "Dawn of the transformer era in speech emotion recognition: Closing the valence gap," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10745–10759, Mar. 2023.

[13] W. X. Zhao et al., "A survey of large language models," 2023, *arXiv:2303.18223*.

[14] T. Brown et al., "Language models are few-shot learners," in *Proc. 34th Annu. Conf. Neural Inf. Process. Syst. (NeuIPS)*, 2020, pp. 1877–1901.

[15] X. Ren et al., "PanGu-σ: Towards trillion parameter language model with sparse heterogeneous computing," 2023, *arXiv:2303.10845*.

[16] J. Hoffmann et al., "Training compute-optimal large language models," 2022, *arXiv:2203.15556*.

[17] L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," *Proc. IEEE*, vol. 108, no. 4, pp. 485–532, Aug. 2020.

[18] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, Dec. 2020.

[19] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, "Efficient acceleration of deep learning inference on resource-constrained edge devices: A review," *Proc. IEEE*, vol. 111, no. 4, pp. 42–91, Jan. 2023.

[20] C. Zhang, X. Yuan, Q. Zhang, G. Zhu, L. Cheng, and N. Zhang, "Toward tailored models on private AIoT devices: Federated direct neural architecture search," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17309–17322, Sep. 2022.

[21] P. Ganesh et al., "Compressing large-scale transformer-based models: A case study on BERT," *Trans. Assoc. Comput. Linguist.*, vol. 9, pp. 1061–1080, Sep. 2021.

[22] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1–28, Dec. 2022.

[23] C. Liu, Z. Zhang, and D. Wang, "Pruning deep neural networks by optimal brain damage," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2014, pp. 1092–1095.

[24] Y. Lin, C. Wang, C. Chang, and H. Sun, "An efficient framework for counting pedestrians crossing a line using low-cost devices: The benefits of distilling the knowledge in a neural network," *Multimedia Tools Appl.*, vol. 80, no. 3, pp. 4037–4051, Sep. 2021.

[25] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," 2020, *arXiv:2006.04768*.

[26] Z. Zhang, T. Farnsworth, S. Lin, and S. Karout, "WakeUpNet: A mobile-transformer based framework for end-to-end streaming voice trigger," 2022, *arXiv:2210.02904*.

[27] J. Han, Z. Zhang, M. Schmitt, M. Pantic, and B. Schuller, "From hard to soft: Towards more human-like emotion recognition by modelling the perception uncertainty," in *Proc. 25th ACM Int. Conf. Multimedia (ACM MM)*, 2017, pp. 890–897.

[28] J. Kaplan et al., "Scaling laws for neural language models," 2020, *arXiv:2001.08361*

[29] T. Henighan et al., "Scaling laws for autoregressive generative modeling," 2020, *arXiv:2010.14701*.

[30] S. Ma, R. Bassily, and M. Belkin, "The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 3331–3340.

[31] A. Nediyanchath, P. Paramasivam, and P. Yenigalla, "Multi-head attention for speech emotion recognition with auxiliary learning of gender recognition," in *Proc. 45th IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 7179–7183.

[32] B. Schuller et al., "The INTERSPEECH 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism," in *Proc. 14th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2013, pp. 148–152.

[33] L. Tarantino, P. N. Garner, and A. Lazaridis, "Self-attention for speech emotion recognition," in *Proc. 20th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2019, pp. 2578–2582.

[34] W. Chen, X. Xing, X. Xu, J. Pang, and L. Du, "SpeechFormer: A hierarchical efficient framework incorporating the characteristics of speech," in *Proc. 23th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2022, pp. 346–350.

[35] S. Chen, X. Xing, W. Zhang, W. Chen, and X. Xu, "DWFormer: Dynamic window transFormer for speech emotion recognition," in *Proc. 48th IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2023, pp. 1–5.

[36] Y. Wang, A. Boumadane, and A. Heba, "A fine-tuned wav2vec 2.0/HuBERT benchmark for speech emotion recognition, speaker verification and spoken language understanding," 2021, *arXiv:2111.02735*.

[37] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?" in *Proc. 33th Annu. Conf. Neural Inf. Process. Syst. (NeuIPS)*, 2019, pp. 14014–14024.

[38] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," in *Proc. 57th Conf. Assoc. Comput. Linguist. (ACL)*, 2019, pp. 5797–5808.

[39] S. Prasanna, A. Rogers, and A. Rumshisky, "When BERT plays the lottery, all tickets are winning," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2020, pp. 3208–3229.

[40] X. Chen, Y. Cheng, S. Wang, Z. Gan, Z. Wang, and J. Liu, "EarlyBERT: Efficient BERT training via early-bird lottery tickets," in *Proc. 59th Conf. Assoc. Comput. Linguist. (ACL)*, 2021, pp. 2195–2207.

[41] A. Fan, E. Grave, and A. Joulin, "Reducing transformer depth on demand with structured dropout," 2019, *arXiv:1909.11556*.

[42] H. Sajjad, F. Dalvi, N. Durrani, and P. Nakov, "On the effect of dropping layers of pre-trained transformer models," *Comput. Speech Lang.*, vol. 77, Jul. 2023, Art. no. 101429.

[43] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

[44] X. Jiao et al., "TinyBERT: Distilling BERT for natural language understanding," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2020, pp. 4163–4174.

[45] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "MobileBERT: A compact task-agnostic BERT for resource-limited devices," in *Proc. 58th Conf. Assoc. Comput. Linguist. (ACL)*, 2020, pp. 2158–2170.

[46] H. Chang, S. Yang, and H. Lee, "DistilHuBERT: Speech representation learning by layer-wise distillation of hidden-unit BERT," in *Proc. 47th IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2022, pp. 7087–7091.

[47] T. M. Nguyen, V. Suliafu, S. J. Osher, L. Chen, and B. Wang, "FMMformer: Efficient and flexible transformer via decomposed near-field and far-field attention," in *Proc. 35th Annu. Conf. Neural Inf. Process. Syst. (NeuIPS)*, 2021, pp. 29449–29463.

[48] Z. Geng, M. Guo, H. Chen, X. Li, K. Wei, and Z. Lin, "Is attention better than matrix decomposition?" in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–24.

[49] P. H. Chen, H. Yu, I. S. Dhillon, and C. Hsieh, "DRONE: Data-aware low-rank compression for large NLP models," in *Proc. 34th Annu. Conf. Neural Inf. Process. Syst. (NeuIPS)*, 2021, pp. 29321–29334.

[50] G. I. Winata, S. Cahyawijaya, Z. Lin, Z. Liu, and P. Fung, "Lightweight and efficient end-to-end speech recognition using low-rank transformer," in *Proc. 45th IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 6144–6148.

[51] L. Ren, Z. Jia, X. Wang, J. Dong, and W. Wang, "A T^2-tensor-aided multiscale transformer for remaining useful life prediction in IIoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 8108–8118, Nov. 2022.

[52] S. Guo, J. M. Alvarez, and M. Salzmann, "ExpandNets: Linear over-parameterization to train compact convolutional networks," in *Proc. 34th Annu. Conf. Neural Inf. Process. Syst. (NeuIPS)*, 2020, pp. 1298–1310.

[53] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," in *Proc. 34th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 13733–13742.
[54] X. Ding, Y. Guo, G. Ding, and J. Han, "ACNet: Strengthening the kernel skeletons for powerful CNN via asymmetric convolution blocks," in *Proc. 32th Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1911–1920.
[55] C. Yeh et al., "Transformer-transducer: End-to-end speech recognition with self-attention," 2019, *arXiv:1910.12977*.
[56] A. Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. 21st Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2020, pp. 5036–5040.
[57] S. Arora, N. Cohen, and E. Hazan, "On the optimization of deep networks: Implicit acceleration by overparameterization," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 244–253.
[58] D. Hernandez, J. Kaplan, T. Henighan, and S. McCandlish, "Scaling laws for transfer," 2021, *arXiv:2102.01293*.
[59] C. Busso et al., "IEMOCAP: Interactive emotional dyadic motion capture database," *Lang. Resour. Eval.*, vol. 42, no. 4, pp. 335–359, Nov. 2008.
[60] J. Zhao et al., "M3ED: Multi-modal multi-scene multi-label emotional dialogue database," in *Proc. 60th Conf. Assoc. Comput. Linguist. (ACL)*, 2022, pp. 5699–5710.
[61] J. Gratch et al., "The distress analysis interview corpus of human and computer interviews," in *Proc. 9th Int. Conf. Lang. Resour. Eval. (LREC)*, 2014, pp. 3123–3128.
[62] L. Guo, L. Wang, J. Dang, Y. Fu, J. Liu, and S. Ding, "Emotion recognition with multimodal transformer fusion framework based on acoustic and lexical information," *IEEE MultiMedia*, vol. 29, no. 2, pp. 94–103, Apr. 2022.
[63] M. Ren, X. Huang, W. Li, D. Song, and W. Nie, "LR-GCN: Latent relation-aware graph convolutional network for conversational emotion recognition," *IEEE Trans. Multimedia*, vol. 24, pp. 4422–4432, Oct. 2021.
[64] M. Hou, Z. Zhang, C. Liu, and G. Lu, "Semantic alignment network for multi-modal emotion recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 5318–5329, Feb. 2023.
[65] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and Generalization in Overparameterized neural networks," in *Proc. 33th Annu. Conf. Neural Inf. Process. Syst. (NeuIPS)*, 2019, pp. 6155–6166.
[66] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 242–252.
[67] L. Guo, L. Wang, C. Xu, J. Dang, E. S. Chng, and H. Li, "Representation learning with spectro-temporal-channel attention for speech emotion recognition," in *Proc. 46th IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2021, pp. 6304–6308.
[68] Z. Huang, J. Epps, and D. Joachim, "Exploiting vocal tract coordination using dilated CNNs for depression detection in naturalistic environments," in *Proc. 45th IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 6549–6553.
[69] A. Othmani, D. Kadoch, K. Bentounes, E. Rejaibi, R. Alfred, and A. Hadid, "Towards robust deep neural networks for affect and depression recognition from speech," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR). ICPR Int. Workshops Chall.*, 2021, pp. 5–19.
[70] A. Saidi, S. B. Othman, and S. B. Saoud, "Hybrid CNN-SVM classifier for efficient depression detection system," in *Proc. 4th Int. Conf. Adv. Syst. Emerg. Technol. (IC_ASET)*, 2020, pp. 229–234.
[71] H. Solieman and E. A. Pustozerov, "The detection of depression using multimodal models based on text and voice quality features," in *Proc. IEEE Conf. Russ. Young Res. Electr. Electron. Eng. (ElConRus)*, 2021, pp. 1843–1848.
[72] S. d'Ascoli, M. Refinetti, G. Biroli, and F. Krzakala, "Double trouble in double descent: Bias and variance(s) in the lazy regime," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 2280–2290.
[73] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: Where bigger models and more data hurt," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–24.

Zixing Zhang (Senior Member, IEEE) received the master's degree in physical electronics from Beijing University of Posts and Telecommunications, Beijing, China, in 2010, and the Ph.D. degree in computer engineering from the Technical University of Munich, Munich, Germany, in 2015.

He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. From 2017 to 2019, he was a Research Associate with the Department of Computing, Imperial College London, London, U.K. Before that, he was a Postdoctoral Researcher with the University of Passau, Passau, Germany. He has authored more than 120 publications in peer-reviewed books, journals, and conference proceedings, leading to more than 5500 citations (H-index 44). His research interests include human-centered emotion and health computation.

Prof. Zhang serves as an Associate Editor for IEEE TRANSACTIONS ON AFFECTIVE COMPUTING and *Frontiers in Signal Processing*, an Editorial Board Member for *Scientific Reports* (Nature), and a Guest Editor for IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE.

Zhongren Dong (Student Member, IEEE) received the master's degree from Zhengzhou University, Zhengzhou, China, in 2021. He is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China.

His research interests include self-supervised learning and model compression for mental health and emotion recognition with speech signals.

Weixiang Xu received the master's degree from Hunan Normal University, Changsha, China, in 2022. He is currently pursing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha.

His research interests include affective computing, multimodal deep learning, and large language models.

Jing Han (Senior Member, IEEE) received the bachelor's degree in electronic and information engineering from Harbin Engineering University, Harbin, China, in 2011, the master's degree from Nanyang Technological University, Singapore, in 2014, and the Ph.D. degree in computer science from the University of Augsburg, Augsburg, Germany, in 2019.

She is currently a Senior Research Associate with the Department of Computer Science and Technology, University of Cambridge, Cambridge, U.K. She (co)-authored more than 60 publications in peer-reviewed journals and conference proceedings. Her research interests include affective computing and digital health.

Dr. Han has served as a Program Committee Member for the Audio/Visual Emotion Challenge and Workshop in 2018, a Technical Program Committee Member for the Association for Computing Machinery Multimedia since 2019, a Leading Guest Editor for IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, and an Associate Editor for IEEE TRANSACTIONS ON AFFECTIVE COMPUTING.