A Knowledge Distillation-based Approach to Speech Emotion Recognition

Ziping Zhao#, *Member, IEEE,* Jixin Liu#, Haishuai Wang, *Member, IEEE,* Danushka Bandara, *Member, IEEE,* and Jianhua Tao, *Senior Member, IEEE*

Abstract—Due to rapid advancements in deep learning, Transformer-based architectures have proven effective in speech emotion recognition (SER), largely due to their ability to model long-term dependencies more effectively than recurrent networks. The current Transformer architecture is not well-suited for SER because its large parameter number demands significant computational resources, making it less feasible in environments with limited resources. Furthermore, its application to SER is limited because human emotions, which are expressed in long segments of continuous speech, are inherently complex and ambiguous. Therefore, designing specialized Transformer models tailored for SER is essential. To address these challenges, we propose a novel knowledge distillation framework that combines meta-knowledge and curriculum-based distillation. Specifically, we fine-tune the teacher model to optimize it for the SER task. For the student model, we embed individual sequence time points into variable tokens, which are used to aggregate the global speech representation. Additionally, we combine supervised contrastive and cross-entropy loss to increase the inter-class distance between learnable features. Finally, we optimize the student model using both meta-knowledge and the curriculum-based distillation framework. Experimental results on two benchmark datasets, IEMOCAP and MELD, demonstrate that our method performs competitively with state-of-the-art approaches in SER.

Index Terms—Contrastive learning, knowledge distillation, speech emotion recognition, transformer.

1 Introduction

SPEECH is a crucial form of human interaction, which conveys important concepts and emotions. Speech emotion recognition (SER) aims to predict the emotional content embedded in speech signals, with applications in fields such as healthcare systems and human-computer interaction [1]. As a result, SER remains an ongoing focus within the research community. The rapid advancement of deep learning has led to the development of numerous advanced models, yielding promising results in SER. Notably, convolutional (CNNs), graph (GNNs), and recurrent neural networks (RNNs) as well as two popular variants—long short-term memory (LSTM) networks and gated recurrent units (GRUs)—have been extensively studied and applied to SER tasks [2]–[6].

Recent advancements in transformer-based architectures have demonstrated significant potential in various artificial intelligence tasks. The transformer [7], as a primary backbone architecture, has gained considerable attention within the natural language processing (NLP) community [8] due to its exceptional performance across various applications. Recently, transformer-based end-to-end models have also shown considerable success in several speech-related fields, such as automatic speech recognition (ASR) [9], [10], and speech enhancement [11] and separation [12]. Additionally, transformers are increasingly being applied to SER due to their capacity to model long-term dependencies in sequences, with several variants proposed to improve performance [13], [14].

Two key challenges render the standard transformer model unsuitable for use in transformer-based SER applications. First, the transformer's computational cost and memory consumption increases linearly with sequence length and quadratically with hidden dimension size, creating challenges for large-scale data training and inference [15], [16]. Currently, available models are too large to be used for edge speech applications on devices with limited resources. To address this challenge, several studies have applied knowledge distillation (KD), a key model compression technique, to derive compact models from larger, more complex ones [17]. The teacher-student model [18] has also been employed to reduce the size and complexity of SER models.

During traditional knowledge distillation, the teacher is unaware of the student's capacity and is not specifically optimized for the distillation process [19]. The teacher model is typically unaware of its requirement to transfer knowledge to the student, often resulting in suboptimal knowledge transfer [19].

To address this, Zhou et al. introduced MetaDistil, a novel teacher-student distillation framework that leverages meta-learning to provide feedback on the student's learning progress, thereby enhancing the teacher's ability to transfer knowledge during the distillation process [19]. While MetaDistil addresses this issue to some degree, a gap persists in the typical teacher-student relationship where both models should ideally reinforce each other. However, the student model often remains less effective than the teacher. Furthermore, the MPDistil model [20] combines meta-policy distillation with curriculum learning, allowing the student model to surpass the teacher.

Second, conventional transformers struggle with forecasting over longer periods due to performance degradation and

This work was supported by the STI 2030-Major Projects (Grant No. 2021ZD0201500), and the National Natural Science Foundation of China (Grant Nos. 62071330, 61831022, U21B2020).
Ziping Zhao is with the College of Computer and Information Engineering, Tianjin Normal University, Tianjin, China. (ztianjin@126.com)
Jixin Liu is with the College of Computer and Information Engineering, Tianjin Normal University, Tianjin, China. (liujixin2@outlook.com)
Haishuai Wang is with the College of Computer Science, Zhejiang University, Zhejiang, China. (haishuai.wang@gmail.com)
Danushka Bandara is with the Fairfield University, Department of Computer Science and Engineering, Fairfield, Connecticut, 06824, USA (danushkabandara@gmail.com)
Jianhua Tao is with the Department of Automation, Tsinghua University, Beijing, China. (jhtao@tsinghua.edu.cn)
 # Both authors contributed equally to this work.
Corresponding author: Jianhua Tao.

computation explosion, particularly when the time series data involves long-range dependencies, such as those in speech.

Meanwhile, for the SER task, speech data typically involves multiple acoustic features (e.g., pitch, energy, and formants), each with its own temporal structure. However, conventional transformers may fail to properly model these features due to unaligned timestamps and distinct physical measurements across the variables, resulting in ineffective attention maps. To address this, iTransformer [21] embeds time points into variate tokens, allowing the attention mechanism to focus on the relationships between different speech features in a multivariate context, enhancing variate-centric representation learning.

Motivated by the above observations, we propose a novel architecture that integrates meta-knowledge and curriculum-based knowledge distillation into SER. We aim to enable the student model to effectively adapt to complex, diverse emotional expressions in speech and improve over time, ensuring better performance and generalization in real-world SER applications. Specifically, we utilize the Hubert-large-ls960-ft model as the teacher and fine-tune it for SER. To address the issue of meaningless mappings in traditional transformer SER models, we embed time points from individual sequences into variable tokens in the student model. This aggregates global representations and uses the attention mechanism to capture multivariate correlations effectively. To further enhance performance, we incorporate both supervised contrastive learning and cross-entropy loss [22]. Furthermore, we perform comprehensive experiments on the IEMOCAP and MELD datasets, with results that demonstrate competitive performance compared to state-of-the-art models. In summary, our contributions are outlined as follows:

- We developed a novel knowledge distillation framework combining meta-knowledge and curriculum-based distillation. To the best of the authors' knowledge, this is the first time that such a hybrid architecture has been employed for SER.
- Inspired by the iTransformer, we introduce it for SER to overcome the limitations mentioned earlier in this study, and we also combine supervised contrastive and cross-entropy loss to further improve the model performance.

2 Related Work

2.1 Speech transformers

Transformers have attracted considerable attention in the speech domain due to their versatility across tasks. Recent studies show that transformer-based methods can significantly improve feature extraction and model performance. For example, Wang et al. [23] proposed an end-to-end speech emotion recognition model using stacked transformer blocks to strengthen global feature representation. However, while this approach captures global information adequately, it may overlook local features, which are also crucial for SER.

To address this limitation, Hu et al. [24] leveraged multiple models, including a residual BLSTM, a CNN, and an E-transformer module, to capture local and global speech information. This approach improves the model's learning capacity and provides a more comprehensive understanding of speech features by integrating multi-scale representations.

Recent advancements in self-supervised transformer models have led to notable success in speech processing tasks. For instance, the wav2vec model family [25], including VQ-wav2vec [26] and wav2vec 2.0 [27], has made significant advances in ASR. These models have also been effectively used for SER through transfer learning. For instance, Pepino et al. [28] employed a pre-trained wav2vec 2.0 model, combining outputs from multiple layers to generate a richer representation of speech features.

Then, Cai et al. [29] proposed a multi-task learning (MTL) framework that uses the wav2vec 2.0 model for feature extraction while jointly training for speech emotion classification and text recognition. This approach enhances feature extraction and highlights the flexibility of transformer-based models in managing multiple tasks simultaneously.

2.2 Knowledge distillation

Knowledge distillation is a crucial technique for training compact networks, enabling them to achieve performance levels similar to deeper, more complex models. This concept was first introduced by Hinton et al. [17], where student models were trained to minimize the discrepancies in the softened class probabilities generated by both teacher and student models. This foundational approach significantly enhanced the similarity between the teacher and student models' outputs. However, this method primarily focused on aligning the output probabilities and did not fully leverage the intermediate representations or attention mechanisms that could further improve knowledge transfer.

Afterwards, Romero et al. [30] introduced a novel strategy by utilizing intermediate representations from the teacher to guide student model training. This approach improved knowledge transfer by ensuring that the student model mimicked the final outputs and learned the hierarchical features captured by the teacher. Further advancements were made by Zagoruyko et al. [31] who significantly boosted performance by incorporating CNN attention mechanisms. This approach guided the student model to mimic the attention map of a more powerful teacher network, enhancing the focus on critical features.

In contrast, Park et al. [32] proposed relational knowledge distillation, a framework that captures the interrelationships between data samples through distance and angle metrics. This innovative approach has inspired several variants [33]–[38], each building on the original strategies to further refine the distillation process.

Despite these advancements, a notable limitation persists: as the gap between the teacher and student models widens, the performance of the student model tends to degrade. This suggests that while a teacher model can effectively transfer knowledge to a student model of a certain size, it becomes less effective when the student model is significantly smaller.

2.3 Meta-knowledge distillation

Meta-knowledge distillation techniques employ meta-learning to enhance the instructional efficiency of teacher models. A notable example is the work by Pan et al. [39] who developed a cross-domain meta-teacher model capable of capturing and transferring knowledge across different domains. This model is constructed by extracting instance- and feature-level information about transferable knowledge from various domains. Then, the guidance provided by the meta-teacher is used to train a single-domain student model, thereby improving its generalizability. While this approach demonstrates the potential of meta-learning in knowledge transfer, it focuses primarily on cross-domain knowledge transfer and may not fully address other challenges, such as the teacher-student gap or data enhancement incompatibility.

To address these limitations, Liu et al. [40] proposed a meta-knowledge distillation approach that introduces learnable

meta-temperature parameters. This method specifically targets the teacher-student gap and resolves data enhancement incompatibility, which are critical issues during practical applications. By dynamically adjusting the temperature parameters, this method improves the knowledge distillation efficiency and broadens its applicability to diverse datasets and models.

Based on these advancements, Zhou et al. [19] developed MetaDistil, a versatile meta-knowledge distillation framework. This framework leverages meta-learning to provide feedback on the student's learning progress, thereby enhancing the teacher's knowledge transfer during the distillation process. Unlike previous methods that focused on cross-domain transfer or teacher-student alignment, MetaDistil further refines the knowledge transfer process, ensuring that the teacher model's guidance is more effective and adaptive to the student's learning trajectory.

It is important to note that the methods mentioned above focus on having the teacher model dynamically guide the student to improve its efficiency, but they do not aim to make the models outperform one another.

2.4 Curriculum-based knowledge distillation

Curriculum-based learning involves progressively improving training samples over the course of the training period [41]. Lao et al. [42] proposed a curriculum learning approach driven by linguistic bias, where two learning phases—easy and hard—are gradually introduced. Then, examples from different phases are selected for learning based on a difficulty metric. Knowledge distillation techniques are also introduced to prevent the model from forgetting previously learned information. This approach enables the model to gradually acquire more complex knowledge by progressively increasing the task difficulty. However, this method's stage division is rather coarse, which may reduce the seamlessness of the model's transition between various stages, potentially affecting the overall learning effect.

To address the issue of insufficiently fine-grained stage delineation, Amara et al. [43] proposed a curriculum-based expert selection method, where a tiered instructional curriculum facilitates faster and more effective learning of complex data samples from a high-capacity teacher network. This method selects a teacher for each input and determines the teacher network's capacity according to the task's classification difficulty. This approach improves the model's learning efficiency for complex data and enhances the flexibility and relevance of knowledge distillation by dynamically selecting teacher models.

Accordingly, Li et al. [44] further proposed a curriculum temperature distillation method based on dynamic temperature parameters. This method makes the distillation process progressively more challenging by introducing dynamic, learnable temperature parameters and adjusting the distillation process according to the task difficulty. Specifically, the correlation between temperature parameters and distillation losses during the process is gradually enhanced, resulting in a progressive increase in difficulty. This approach optimizes the dynamics of the distillation process and further enhances the overall knowledge distillation framework.

Next, Sengupta et al. [20] proposed a collaborative and competitive optimization framework based on the above approach. The framework combines a meta-knowledge distillation paradigm, introduces collaborative and competitive utility optimization procedures, and designs a task-oriented curriculum-based reward system for student models. Through this collaborative and competitive mechanism, the instructor and student models are encouraged to outperform one another, further enhancing the overall learning outcomes.

3 Methodology

This section provides a brief overview of the four training steps of the model: fine-tuning the teacher model, distilling knowledge from the teacher to the student, meta-teacher learning, and student curriculum learning. The overall architecture of the proposed model is illustrated in Fig. 1.

3.1 Fine-tuning the teacher model

At this stage, we used the Hubert-large-ls960-ft as the teacher model, as described in [45]. The pre-trained weights from this model were used as initial weights for the teacher encoder and transferred to the classifier for fine-tuning during downstream tasks. For a given training sample batch of size N, denoted as D_{train} = (x_1, y_1), . . . ,(x_N, y_N), we computed the loss and applied gradient descent to update the parameters of the teacher model, θ_T. Therefore, the teacher model's loss is defined as:

$$L^{teacher} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{y}(i,T)), \tag{1}$$

where $\hat{y}(i,T)$ represents the output generated by the teacher model T with parameter θ_T, y_i is the label of the i-th training sample, and L denotes the cross-entropy loss.

3.2 Distilling knowledge from the teacher to the student

This subsection starts with a detailed description of our proposed student model, followed by an explanation of its training process.

3.2.1 Student model

The student model consists of multiple encoder layers, and we briefly describe the function of each component. Initially, the individual sequences' time points were embedded as variable tokens. Next, an attention mechanism was employed to capture multivariate correlations. Then, we applied normalization to the cascade representation of individual variables to handle non-stationarity.

At the same time, discrepancies caused by inconsistent measurements were mitigated by normalizing all the series (or variable) tokens to a standard normal distribution. Then, we learned the nonlinear representation by applying a feed-forward network to each variable token. Finally, an average pooling layer was applied, and the output was fed into the classifier to extract sentiment recognition information.

Simultaneously, we combined supervised contrastive learning and cross-entropy loss to address boundary ambiguity issues. For contrastive learning, we employed MoCo [46] as the framework, and for data augmentation, we used SpecAugment [47], which operates on the input audio's features instead of the raw audio. The overall structure of the student model is illustrated in the lower-left corner of Fig. 1. The student model's loss function is defined as follows:

$$L^{student} = \lambda L^{sup} + (1-\lambda)\frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{y}(i,S)), \tag{2}$$

where (2) includes a supervised contrastive L^{sup} and a cross-entropy loss L. In this instance, λ is a hyperparameter that requires tuning during training. For a given batch of N training

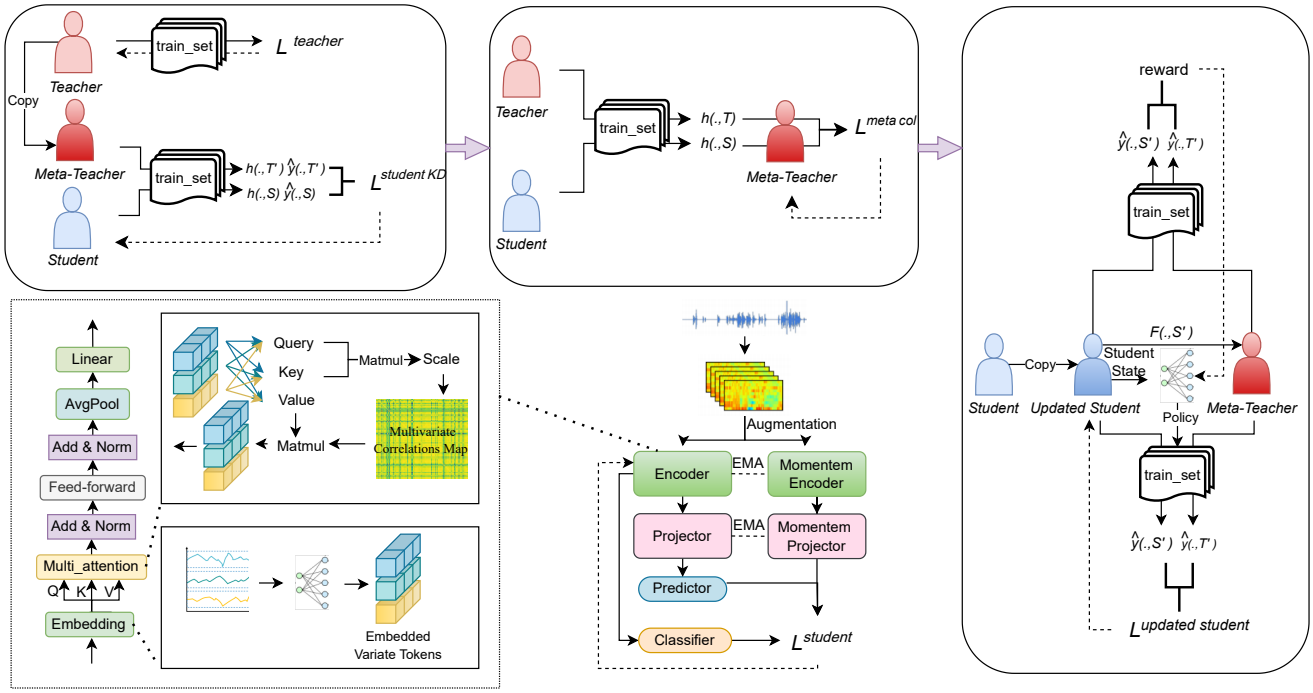


Fig. 1: Illustrating the overall model structure through multiple figures, including fine-tuning the teacher and training the student model, the meta-learning process, the student curriculum learning phase, and the overall student model structure, which has a feed-forward and normalization structure similar to the transformer but shifts the focus from the time to the variable dimension.

samples, the cross-entropy loss is calculated using the true label y_i and the predicted output $\hat{y}(i, S)$ generated by the student model S with the parameter θ_S.

Thus, the supervised contrastive loss is computed as:

$$L^{sup} = \sum_{k \in K} \frac{-1}{|P(k)|} \sum_{p \in P(k)} log \frac{exp(z_k \cdot z_p / \tau_1)}{\sum_{a \in A(k)} exp(z_k \cdot z_a / \tau_1)}, \quad (3)$$

where K is the set of all samples and k serves as an anchor sample. In (3), the "$\cdot$" symbols denote the inner product, τ_1 is a scalar temperature parameter, and $A(k)$ represents the set of all samples except the anchor sample k. $P(k) \equiv \{p \in A(k) : \tilde{y}_p = \tilde{y}_k\}$ is a collection of all the positives that are different from k, the cardinality $|P(k)|$ represents the number of such positive samples, and z_k is the normalized embedding of the anchor sample k. In addition, z_p is the normalized embedding of a positive sample from the set that contains all the samples in the batch that share the same label as the anchor sample k, and z_a is the normalized embedding of a sample a in the set of all samples except the anchor.

3.2.2 Training the student model

In the second step, we distilled the student model S using knowledge from the meta-teacher, which is a replica of the fine-tuned teacher model described in the next subsection. Then, the training sample was fed into the student and meta-teacher models to extract encoder-generated hidden representations $h(., S)$, $h(., T')$ and predicted classifications $\hat{y}(., S)$, $\hat{y}(., T')$.

Following the method proposed by Zhou et al. [19], the student model's parameters were updated according to the loss function:

$$L^{student\ KD} = \alpha L^{student} + (1 - \alpha) L^{soft} + \beta L^{PKD}, \quad (4)$$

which combines three components: the student model's loss function $L^{student}$(as defined in (2)), the Kullback-Leibler divergence between the meta-teacher and student logits L^{soft}, and the mean-squared error between the meta-teacher and student hidden representations L^{PKD}.

Moreover, L^{soft} measures the divergence between the student's and meta-teacher's probability distributions:

$$L^{soft} = \tau_2^2 \sum_{i=1}^{N} p_s(i) ln(\frac{p_s(i)}{p_t(i)}), \quad (5)$$

where the temperature parameter τ_2 is a positive scalar that softens the output probabilities of the student and meta-teacher models. Specifically, the probabilities p_s and p_t are computed using:

$$p_s(i) = \phi_1(\frac{\hat{y}(i, S)}{\tau_2}), \quad (6)$$

$$p_t(i) = \phi_2(\frac{\hat{y}(i, T')}{\tau_2}), \quad (7)$$

where ϕ_1 denotes the log-softmax activation function, and ϕ_2 represents the softmax activation function, which normalizes a logits vector into a probability distribution. Additionally, $\hat{y}(i, S)$ and $\hat{y}(i, T')$ denote the raw logits (i. e., the unnormalized scores) output from the student S and the meta-teacher model T' for class i, respectively. Scaling the logits using the temperature parameter τ_2 before applying the softmax produces a softer probability distribution, which is useful for knowledge distillation as it emphasizes the meta-teacher's class similarity knowledge. The hyperparameters α, β, and τ_2 also require manual adjustment.

The PKD loss term L^{PKD} measures the discrepancy between the meta-teacher and student networks' hidden representations:

$$L^{PKD} = \|h(.,T') - h(.,S)\|_2, \tag{8}$$

where $\|.\|_2$ denotes the L2 norm, which quantifies the distance between the meta-teacher's and student's feature representations, with a smaller value indicating better alignment. In this instance, $h(.,T')$ is the feature representation produced by the meta-teacher model T' for a given input, and $h(.,S)$ is the feature representation produced by the student model S for the same input.

It is worth noting that other distillation strategies can also be used.

3.3 Meta-teacher learning

The meta-teacher model, denoted as T' with parameter $\theta_{T'}$, uses a simple feed-forward network architecture to generate final outputs from the teacher and student models' hidden representations, effectively acting as a classifier. We created a meta-teacher by cloning the fine-tuned teacher model. During the meta-teacher training, the encoder parameters were frozen while the feed-forward network parameters were updated according to the loss function.

To enable the teacher to focus on the student model's performance, we passed the training sample through both models to obtain the tenderized intermediate features $h(i,T)$ and $h(i,S)$, respectively. Then, we passed the intermediate features through the meta-teacher model to obtain the outputs $\hat{\hat{y}}(i,T) = T'(h(i,T),\theta_{T'})$ and $\hat{\hat{y}}(i,S) = T'(h(i,S),\theta_{T'})$. Adhering to Sengupta's method [20], we used only the true class's probabilities $\bar{y}(i,T) = \phi(\hat{\hat{y}}(i,T))_c$ and $\bar{y}(i,S) = \phi(\hat{\hat{y}}(i,S))_c$ for $c = argmax_k\ y(i,k)$, the true class label. Hence, the loss function is defined as:

$$L^{meta\ col} = -\frac{1}{2N}\sum_{i=1}^{N}[log\ \bar{y}(i,T) + log\ \bar{y}(i,S)]. \tag{9}$$

It is crucial to highlight that the student and meta-teacher learning models are optimized simultaneously for dynamic instruction.

3.4 Student curriculum learning

This section focuses on enabling the student model to outperform the teacher across various tasks. To help the student learn the different task characteristics, we employed a curriculum model. The model is a feed-forward network that captures the student model's current state and samples a single task (or action) during each forward pass.

3.4.1 Training the updated student model

Similar to the meta-teacher training, the updated student model is denoted as S' with parameters $\theta_{S'}$. We cloned the student model into the retrained system. The dataset was sampled following the curriculum model outlined in the next subsection. Then, the selected training samples were fed into the updated student model to generate the predicted classifications $\hat{y}(i,S')$, the student state, and the intermediate features $F(i,S')$. These intermediate features $F(i,S')$ were subsequently fed into the frozen meta-teacher model to obtain the predicted classifications $\hat{y}(i,T')$. The updated student model parameters $\theta_{S'}$ are determined by minimizing the composite loss function:

$$L^{updated\ student} = L^{student'} + L^{mean}, \tag{10}$$

which consists of two essential components: the refined student $L^{student'}$ and the mean consistency loss L^{mean}.

In this instance, $L^{student'}$ is formulated as a hybrid loss function that combines cross-entropy and supervised contrastive loss, capturing both classification accuracy and feature consistency, defined as follows:

$$L^{student'} = \lambda L^{sup} + (1-\lambda)\frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{y}(i,S')), \tag{11}$$

where λ denotes the trade-off parameter balancing the two loss terms.

In addition, the L^{mean} includes a consistency mechanism that uses feedback from the student model, passed through the meta-teacher, to generate predictions. By minimizing the differences between the student and meta-teacher predictions, this method improves feature alignment, resulting in greater model stability and better generalization, defined as:

$$L^{mean} = \frac{1}{N}\sum_{i=1}^{N}[(\hat{y}(i,T')\ -\ \hat{y}(i,S')]. \tag{12}$$

Next, the state was input into the curriculum model to obtain the next dataset tuning classification, highlighting the commonalities across different tasks.

During this phase, we gradually enhanced the quality and effectiveness of the training samples. Specifically, the curriculum model initially focuses on samples with intense emotional expressions, which are easier to recognize. As training progresses, we gradually introduced more samples with subtle or complex emotional expressions to enhance the model's recognition capabilities. This approach allows the model to accumulate emotional expressions knowledge over time, enabling it to better adapt to the challenges of dynamic and complex emotion recognition.

3.4.2 Training the curriculum model

During this phase, the updated student and meta-teacher models are frozen, and they both process the training sample to obtain their predicted classifications $\hat{y}(i,S')$ and $\hat{y}(i,T')$.

To enhance the student model's effectiveness beyond that of the teacher's, the Monte Carlo policy gradient algorithm [48], [49] was employed to update the curriculum model parameters and optimize the reward using the formula provided below:

$$reward = \amalg_{\hat{y}(i,S') > \hat{y}(i,T')}. \tag{13}$$

4 Experiment

We validated our proposed method using the IEMOCAP and MELD datasets. This section begins with an introduction to the datasets and the audio processing operations, followed by a description of our experimental setup and validation strategy.

4.1 Datasets

4.1.1 Interactive emotional dyadic motion capture

The IEMOCAP dataset is a well-known corpus containing audio-visual data recordings and transcriptions of dialogues between two actors. To be consistent with prior works [50], [51], we conducted experiments using a subset of four emotions: angry, happy, sad, and neutral, where the original happy and excited categories are merged into the happy category. The final number of instances for each emotion class is presented in Table 1.

TABLE 1: Instance distribution over four emotion classes–neutral, happy, sad, and angry–for the IEMOCAP dataset.

Session	Neutral	Happy	Sad	Angry	Total
1	384	278	194	229	1,085
2	362	327	197	137	1,023
3	320	286	305	240	1,151
4	258	303	143	327	1,031
5	384	442	245	170	1,241
Sum	1,708	1,636	1,084	1,103	5,531

TABLE 2: Instance distribution over seven emotion classes–**a**nger, **d**isgust, **f**ear, **j**oy, **n**eutral, **s**adness, and **sur**prise–for MELD dataset.

	A.	**D.**	**F.**	**J.**	**N.**	**S.**	**Sur.**	Total
Train	1,109	271	268	1,743	4,710	683	1,205	9,989
Dev	153	22	40	163	470	111	150	1,109
Test	345	68	50	402	1,256	208	281	2,610
Sum	1,607	361	358	2,308	6,436	102	1,636	13,708

4.1.2 Multimodal emotionlines dataset

The MELD dataset is also a benchmark commonly utilized in multimodal emotion recognition during conversations research, which consists of 13.7 hours of dynamic dialogue and contains various scenes from the "friends" TV series. The MELD benchmark dataset contains 7 discrete emotion categories: anger, disgust, sadness, joy, neutral, surprise, and fear. The final number of instances for each emotion class is presented in Table 2.

4.2 Evaluation metrics

Standard evaluation criteria were used to evaluate the results generated by the two datasets. For the IEMOCAP-generated results, we used unweighted accuracy (UA), as defined in (14), and weighted accuracy (WA), as defined in (15), to compare our model to other state-of-the-art methods. Thus, UA and WA are defined as:

$$UA = \frac{\sum_1^C \frac{k_i}{T_i}}{C}, \tag{14}$$

$$WA = \frac{\sum_1^C k_i}{\sum_1^C T_i}, \tag{15}$$

where T_i denotes the number of utterances in the i-th class, k_i represents the number of correctly identified utterances in the i-th class, and C signifies the number of emotional classes.

For the MELD dataset, the official evaluation metric is the weighted F1 score (wF1), which calculates the F1 score for each class, weighted by the number of samples for each label [52].

4.3 Feature extraction

In this study, the Hubert feature was used as the model's input, enabling the extraction of more detailed speech features within the time domain. Hubert is a specialized representation that focuses on encoding temporal information from speech signals. It employs contrastive learning to create meaningful representations by predicting sections of the audio based on other parts of the same segment. Additionally, it utilizes large volumes of unlabeled speech data to produce strong embeddings that capture core speech characteristics.

TABLE 3: Experimental hyperparameter settings.

Local/Global	Params	Datasets	
		IEMOCAP	MELD
Teacher	$learning\ rate$	10^{-4}	10^{-4}
	$\#epochs$	20	20
Student	$learning\ rate$	10^{-4}	10^{-4}
	$\#epochs$	30	100
Meta-teacher	$learning\ rate$	10^{-3}	10^{-3}
	$\#epochs$	30	100
Updated student	$learning\ rate$	10^{-3}	10^{-3}
	$\#epochs$	100	100
Global parameters	batch size	32	32
	α	0.1	0.5
	β	100	100
	λ	0.05	0.2
	τ_1	0.07	1.0
	τ_2	5.0	6.0

Initially, the audio data was processed using the pre-trained model Hubert-large-ll60k [45] to extract features from the IEMOCAP and MELD datasets. Then, the audio segments were adjusted to the appropriate length by filling shorter segments and truncating longer ones, ensuring uniform feature dimensions across the datasets. During this step, each frame was transformed into a frequency domain with a length of 1,024, and a time dimension of 326 for IEMOCAP and 224 for MELD. Consequently, feature maps of size 326 $\times$ 1,024 were generated for IEMOCAP, and 224 $\times$ 1,024 for MELD.

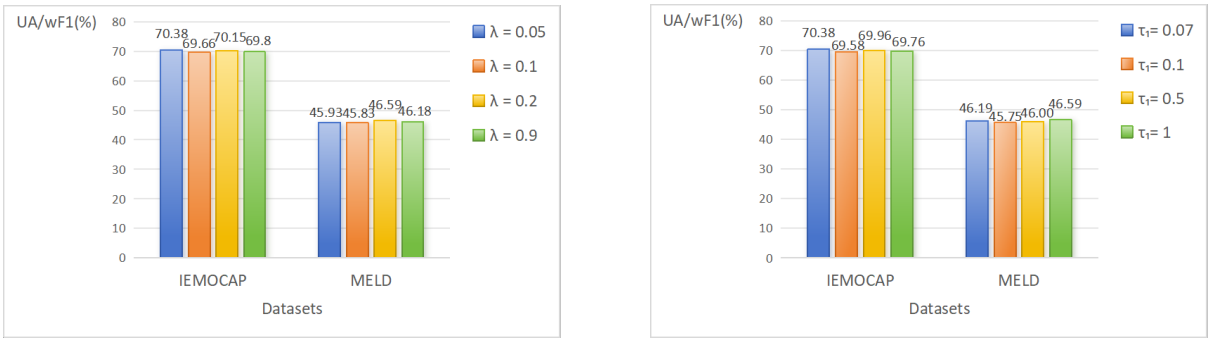
4.4 Experimental details

The experiment began with the construction of the teacher and student models.

The teacher model's initial encoder weights were obtained from the Hubert-large-ls960-ft model. After that, the output from the encoder was transferred to the average pooling layer and classifier for fine-tuning.

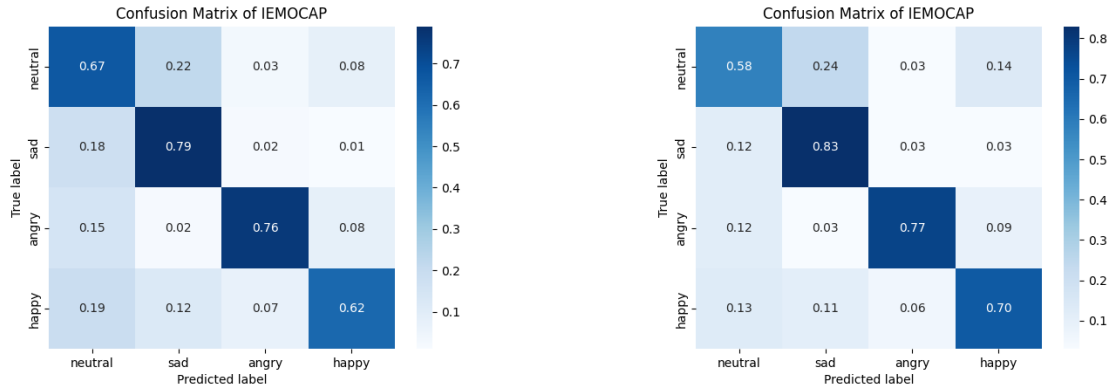
The student model was built using only four encoder layers and consists of two main parts. First, we inverted the data to independently embed the original series of variables into tokens. This was done by reshaping the extracted features from (B, T, N) to (B, N, T), followed by encoding the tokens, with the final shape being (B, N, D). In this instance, B represents the training batch size, T is the length of the time series, N is the dimension of the extracted features, and D is the encoding dimension. Within the encoder, we leveraged the attention mechanism to capture multivariate correlations, applied layer normalization to minimize variable differences, and used the feed-forward network to learn each variable's nonlinear representation.

Additionally, we utilized the MoCo framework [46], which facilitates training using momentum contrast. Within this framework, the encoder is trained using loss backpropagation, while the momentum encoder's parameters are updated as an exponential moving average of the encoder's parameters. The momentum encoder also dynamically constructs the representation dictionary. Simultaneously, the extracted features were employed to mask time-frequency information, with the number of masked frequencies and time steps randomly selected from $[0, 50]$ and $[0, 8]$, respectively. Thus, the encoder processes the extracted features, and the masked features are input into the momentum encoder. The objective is to align the



(a) (b)

Fig. 2: Impact of various hyperparameters on the student model, including (a) λ and (b) τ_1, evaluated using UA and wF1 for the IEMOCAP and MELD datasets, respectively.



(a) (b)

Fig. 3: Confusion matrices for speech emotion recognition on the IEMOCAP dataset depicted in two figures: (a) using cross-entropy loss with time series inversion, and (b) combining cross-entropy and supervised contrastive learning loss.

momentum encoder's output with the representation generated by the encoder network using a contrastive loss.

Similarly, the encoder's output was transferred to the average pooling layer and classifier to produce the corresponding features and classification results.

Next, we applied the training strategy outlined in Section 3.1 to fine-tune the teacher model using the IEMOCAP and MELD datasets for SER. The trained teacher guided the student model, with continuous feedback enabling dynamic instruction. Subsequently, the curriculum learning strategy was implemented to progressively introduce improved datasets, optimizing the student model relative to the teacher.

To conduct comparisons with other state-of-the-art SER models on the IEMOCAP dataset, we also employed the speaker-independent leave_one_session_out (LOSO) method. In this experiment, the model was optimized using the AdamW optimizer. For the student model, we tuned the hyperparameter λ within the range $\{0.05, 0.1, 0.2, 0.9\}$ and the temperature parameter τ_1 in the supervised comparison loss within $\{0.07, 0.1, 0.5, 1.0\}$. For the student knowledge distillation model, we adjusted the hyperparameters τ_2 in the range $\{4.0, 5.0, 6.0\}$, α in $\{0.1, 0.2, 0.5, 0.9\}$, and β in $\{80, 90, 100\}$. The detailed hyperparameter settings are presented in Table 3.

4.5 Results and discussion

4.5.1 Effects of λ and τ_1 on the student model

We adjust the control λ and temperature hyperparameter τ_1 in the student model loss according to the above tuning range. The results are presented in Fig.2(a) and Fig.2(b). The optimal values of λ for IEMOCAP and MELD are 0.05 and 0.2, respectively, and the those for τ_1 are 0.07 and 1, respectively.

Generally, λ is a weighting term that controls the balance between the cross-entropy and supervised contrastive loss. The experimental results typically vary depending on the value of λ. Additionally, while supervised contrastive learning recommends a default value of 0.07 for τ_1, performance can differ slightly across models. Overall, it can be observed that $\tau_1 = 0.07$ is generally a good choice (Fig. 2).

4.5.2 Ablation study

This section provides a detailed discussion of the student model ablation experiments and outlines the knowledge distillation framework.

Table 4 presents the results of the student model ablation study. In this case, (-) inverted refers to the student model without incorporating time points from individual sequences into the variable tokens, while (+) inverted represents the model with the time-aware embedding mechanism applied. The results demonstrate an improved performance with the

TABLE 4: Ablation student model study's UA on the IEMOCAP dataset and wF1 on MELD (%).

Methods	loss	IEMOCAP	MELD
(-) inverted	CE	68.89	42.63
(+) inverted	CE	69.69	45.92
(+) inverted	CE + Sup	**70.38**	**46.59**

TABLE 5: Proposed method's experimental results on the IEMOCAP dataset (%).

Methods	WA	UA
hubert-large-ls960-ft	73.40	74.47
Student	69.41	70.38
Meta-teacher	**74.01**	**75.21**
Distilled student	69.87	70.74
Updated student	**70.25**	**71.27**

TABLE 6: Proposed method's experimental results on the MELD dataset (%).

Methods	WA	wF1
Hubert-large-ls960-ft	51.77	47.06
Student	**52.43**	46.59
Meta-teacher	**52.04**	**48.58**
Distilled student	51.93	46.80
Updated student	52.28	**46.96**

TABLE 7: Comparison of teacher and student model parameters on the IEMOCAP and MELD datasets.

Model	IEMOCAP	MELD
Teacher (Hubert-large-ls960-ft)	302.45M	301.83M
Student	11.92M	11.87M

use of cross-entropy loss, yielding a 0.8 % increase in UA on the IEMOCAP dataset and a 3.29 % improvement in wF1 on MELD compared to the uninverted model. This improvement underscores the model's ability to handle time series data, which often poses challenges for traditional models due to long-term dependencies that can cause unstable predictions. Unlike the iTransformer [21], we removed the projection layer and employed a multilayered classifier to enhance recognition efficiency.

According to Table 4, we observed that our student model achieves a UA of 70.38 % on the IEMOCAP dataset and a wF1 of 46.59 % on MELD when combining the cross-entropy and supervised contrastive losses, with improvements of 0.69 % and 0.67 %, respectively. Figure 3 displays the SER confusion matrix on the IEMOCAP dataset. As shown in Fig. 3(a), the model using only cross-entropy loss achieved moderate performance. However, when supervised contrastive loss was added, as shown in Fig. 3(b), the model showed an improved recognition performance across most categories except for the neutral state. This indicates that incorporating supervised contrastive learning loss enables the model to capture the boundaries and distinguishing features of different states effectively. By addressing boundary ambiguity, this method boosts the recognition accuracy of non-neutral states, especially when handling complex and variable data.

Tables 5 and 6 summarize the performance of different models on the IEMOCAP and MELD datasets, including the fine-tuned teacher, student, meta-teacher, distilled student, and updated student models.

As shown in Table 5, the fine-tuned teacher model achieves a WA of 73.40 % and an UA of 74.47 % on the IEMOCAP dataset, while the student attains a 69.41 % WA and a 70.38 % UA. Similarly, Table 6 indicates that the fine-tuned teacher model achieves 51.77 % WA and a wF1 of 47.06 % on the MELD dataset, while the student attains a 52.43 % WA and a 46.59 % wF1. The parameter statistics in Table 7 show that the teacher model has 25 times more parameters than the student, with 24 encoder layers compared to the student's four. Notably, these statistics only account for the parameters used after feature extraction and exclude those involved in the feature extraction process. Despite this substantial reduction in model size, the student maintains a performance level comparable to the teacher model, highlighting the efficiency and robustness of the proposed lightweight architecture.

Additionally, the meta-teacher model improved performance by 0.61 % in WA and 0.74 % in UA on the IEMOCAP dataset (see Table 5), and by 0.27 % in WA and 1.52 % in wF1 on MELD (see Table 6) compared to the fine-tuned teacher model. These results highlight the effectiveness of the meta-learning approach in strengthening the teacher model's ability to provide dynamic guidance. Furthermore, without curriculum learning, the distilled student outperforms the student model on the IEMOCAP dataset, achieving a 0.46 % higher WA and a 0.36 % higher UA (see Table 5). Conversely, on the MELD dataset, the distilled student achieves a 0.5 % lower WA but 0.21 % higher wF1 compared to the student model (see Table 6). These results underscore the essence of knowledge distillation, which seeks to align the student model's representations with those of the teacher. However, as shown in Table 6, the fine-tuned teacher model performs worse than the student on the MELD dataset, indicating that the teacher's representations may not fully capture the most relevant features for this specific task.

Similarly, Table 5 demonstrates that our proposed method achieved a WA of 70.25 % and a UA of 71.27 % on the IEMOCAP database, which are 0.38 % and 0.53 % higher, respectively, than when the curriculum learning strategy was not applied. Table 6 shows that applying the curriculum learning strategy resulted in a wF1 of 46.96 % on the MELD dataset, improving by 0.16 % compared to when the strategy was not used. This indicates that the strategy aids in updating the student model through meta-teacher learning.

The experimental results presented in Tables 5 and 6 demonstrate that the student model underperforms compared to its teacher counterpart. This performance gap can be primarily attributed to the significantly larger number of parameters in the teacher model. The student model, while optimizing memory usage, sacrifices some valuable parameters in the process.

4.5.3 Comparison with state-of-the-art methods

The baseline models in our experiments are as follows: (i) CA-MSER [50], which extracts multi-level acoustic features and combines them as multi-modal inputs for recognizing subjective human emotions in audio. (ii) TIM-Net [51], a bidirectional, multi-scale network that learns contextual emotion representations across different time scales. (iii) DWFormer [53], which captures temporal significance by dividing samples into windows and incorporating cross-window interaction information for global communication. (iv) ShiftCNN [54], which uses a

TABLE 8: Comparison of the proposed method with the baselines on the IEMOCAP dataset (%).

Methods	Params	WA	UA
CA-MSER [50]	166.53M	69.80	71.05
TIM-Net [51]	0.12M	68.20	69.41
DWFormer [53]	35.71M	69.63	70.69
ShiftCNN [54]	99.03M	**70.40**	**71.31**
SpeechFormer++ [55]	63.76M	69.68	71.09
Ours	11.92M	70.25	71.27

TABLE 9: Comparison of the proposed method with the baselines on the MELD dataset (%).

Methods	Params	WA	wF1
MM-DFN [56]	1.97M	47.70	45.80
DWFormer [53]	25.70M	52.16	46.61
SpeechFormer++ [55]	63.76M	52.20	46.86
Ours	11.87M	**52.28**	**46.96**

time-shift module to mix channel information without introducing additional parameters. (v) SpeechFormer++ [55], an encoder that efficiently models inter- and intra-unit relationships within speech signals, utilizing merge blocks to generate features at various granularities and incorporating a word encoder to integrate word-level features into each unit encoder. This effectively balances fine- and coarse-grained information. (vi) MM-DFN [56], a novel multi-modal dynamic fusion network that captures dynamic changes in contextual information across different semantic spaces by designing a graph-based fusion module to reduce redundant information and enhance modal complementarity.

Tables 8 and 9 compare the proposed method and the baseline models on the IEMOCAP and MELD datasets, respectively. To ensure a fair comparison, we reproduced the methods using the same validation strategy. Our proposed model achieved a WA of 70.25 % and a UA of 71.27 % on the IEMOCAP dataset (see Table 8), and a WA of 52.28 %, and a wF1 of 46.96 % on MELD (see Table 9). This demonstrates that our model outperforms the baselines in results with fewer training parameters, excluding ShiftCNN. The best results for WA and UA are slightly lower than those reported in [54] on the IEMOCAP dataset, as achieving higher predictive performance requires significantly more model parameters and greater computational resources.

5 Conclusion

In this article, we proposed a novel knowledge framework that integrates meta-knowledge and curriculum-based distillation to optimize the teacher model's effectiveness. To address the limitations of traditional transformer models, we incorporated time points from individual sequences as variable SER tokens, drawing inspiration from the iTransformer concept. Extensive experiments on the IEMOCAP and MELD datasets demonstrated that our proposed method outperforms existing approaches.

In future work, we aim to extend the application of our model to other speech-related tasks. We also plan to improve the student model's efficiency—despite having fewer training parameters—so it can match or exceed the more complex teacher model's performance by minimizing the training of non-essential parameters.

References

[1] A. Mohanta and U. Sharma, "Detection of human emotion from speech—tools and techniques," in *Proc. 1st Speech and Language Processing for Human-Machine Communications*, Singapore, 2018, pp. 179–186.

[2] A. Satt, S. Rozenberg, and R. Hoory, "Efficient emotion recognition from speech using deep learning on spectrograms," in *Proc. INTERSPEECH*, Stockholm, Sweden, 2017, pp. 1089–1093.

[3] A. Aftab, A. Morsali, S. Ghaemmaghami, and B. Champagne, "Light-sernet: A lightweight fully convolutional neural network for speech emotion recognition," in *Proc. 45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Singapore, 2022, pp. 6912–6916.

[4] D. Li, J. Liu, Z. Yang, L. Sun, and Z. Wang, "Speech emotion recognition using recurrent neural networks with directional self-attention," *Expert Systems with Applications*, vol. 173, p. 114683, Jul. 2021.

[5] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1d 2d cnn lstm networks," *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, Jan. 2019.

[6] Z. Zhu, W. Dai, Y. Hu, and J. Li, "Speech emotion recognition model based on bi-gru and focal loss," *Pattern Recognition Letters*, vol. 140, pp. 358–365, Dec. 2020.

[7] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[8] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, and L. S. Chao, "Learning deep transformer models for machine translation," in *Proc. 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, 2019, pp. 1810–1822.

[9] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Calgary, Alberta, Canada, 2018, pp. 5884–5888.

[10] Q. Song, B. Sun, and S. Li, "Multimodal sparse transformer network for audio-visual speech recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10 028–10 038, Apr. 2022.

[11] F. Dang, H. Chen, and P. Zhang, "Dpt-fsnet: Dual-path transformer based full-band and sub-band fusion network for speech enhancement," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore, 2022, pp. 6857–6861.

[12] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 2021, pp. 21–25.

[13] W. Chen, X. Xing, X. Xu, J. Pang, and L. Du, "Speechformer: A hierarchical efficient framework incorporating the characteristics of speech," *arXiv preprint arXiv:2203.03812*, vol. 363, no. 1493, pp. 917–921, Mar. 2022.

[14] F. Andayani, L. B. Theng, M. T. Tsun, and C. Chua, "Hybrid lstm-transformer model for emotion recognition from speech audio files," *IEEE Access*, vol. 10, pp. 36 018–36 027, Mar. 2022.

[15] C. Wu, Z. Xiu, Y. Shi, O. Kalinli, C. Fuegen, T. Koehler, and Q. He, "Transformer-based acoustic modeling for streaming speech synthesis," in *Proc. INTERSPEECH*, Brno, Czechia, 2021, pp. 146–150.

[16] S. Latif, A. Zaidi, H. Cuayahuitl, F. Shamshad, M. Shoukat, and J. Qadir, "Transformers in speech processing: A survey," *arXiv preprint arXiv:2303.11607*, 2023.

[17] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, Mar. 2015.

[18] Y. Liu, H. Sun, G. Chen, Q. Wang, Z. Zhao, X. Lu, and L. Wang, "Multi-level knowledge distillation for speech emotion recognition in noisy conditions," in *Proc. INTERSPEECH*, Dublin, Ireland, 2023.

[19] W. Zhou, C. Xu, and J. McAuley, "Bert learns to teach: Knowledge distillation with meta learning," in *Proc. 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, Dublin, Ireland, 2022, pp. 7037–7049.

[20] A. Sengupta, S. Dixit, M. S. Akhtar, and T. Chakraborty, "A good learner can teach better: Teacher-student collaborative knowledge distillation," in *Proc. 12th International Conference on Learning Representations (ICLR)*, Vienna, Austria, 2024.

[21] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[22] X. Wang, S. Zhao, and Y. Qin, "Supervised contrastive learning with nearest neighbor search for speech emotion recognition," in *Proc. INTERSPEECH*, Dublin, Ireland, 2023.
[23] X. Wang, M. Wang, W. Qi, W. Su, X. Wang, and H. Zhou, "A novel end-to-end speech emotion recognition network with stacked transformer layers," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Ontario, Canada, 2021, pp. 6289–6293.
[24] D. Hu, X. Hu, and X. Xu, "Multiple enhancements to lstm for learning emotion-salient features in speech emotion recognition," in *Proc. INTERSPEECH*, Incheon, Korea, 2022, pp. 4720–4724.
[25] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.
[26] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.
[27] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.
[28] L. Pepino, P. Riera, and L. Ferrer, "Emotion recognition from speech using wav2vec 2.0 embeddings," *arXiv preprint arXiv:2104.03502*, 2021.
[29] X. Cai, J. Yuan, R. Zheng, L. Huang, and K. Church, "Speech emotion recognition with multi-task learning." in *Proc. Interspeech*, vol. 2021, Brno, Czech Republic, 2021, pp. 4508–4512.
[30] A. Romero, N. Ballas, E. S. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *Proc. 2nd International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014.
[31] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.
[32] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proc. 32th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, CA, USA, 2019, pp. 3967–3976.
[33] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *Proc. 32th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, CA, USA, 2019, pp. 9163–9171.
[34] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proc. 17th IEEE International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 2019, pp. 1365–1374.
[35] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu *et al.*, "Correlation congruence for knowledge distillation," *Proc. 17th IEEE International Conference on Computer Vision (ICCV)*, pp. 5006–5015, 2019.
[36] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li *et al.*, "Tinybert: Distilling bert for natural language understanding," in *Proc. 25th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Virtual Event, 2020, pp. 4163–4174.
[37] G. Xu, Z. Liu, X. Li, and C. C. Loy, "Knowledge distillation meets self-supervision," in *Proc. 26th European Conference on Computer Vision (ECCV)*, Glasgow, UK, 2020, pp. 588–604.
[38] N. Passalis, M. Tzelepi, and A. Tefas, "Probabilistic knowledge transfer for lightweight deep representation learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 2030–2039, May. 2021.
[39] H. Pan, C. Wang, M. Qiu, Y. Zhang, Y. Li, and J. Huang, "Meta-kd: A meta knowledge distillation framework for language model compression across domains," *arXiv preprint arXiv:2012.01266*, pp. 3026–3036, Nov. 2021.
[40] J. Liu, B. Liu, H. Li, and Y. Liu, "Meta knowledge distillation," *arXiv preprint arXiv:2202.07940*, Feb. 2022.
[41] L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo, "The fifth pascal recognizing textual entailment challenge," in *Proc. 2nd Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA, 2009.
[42] M. Lao, Y. Guo, Y. Liu, W. Chen, N. Pu, and M. S. Lew, "From superficial to deep: Language bias driven curriculum learning for visual question answering," in *Proc. 29th ACM International Conference on Multimedia (MM)*, Virtual Event, China, 2021, pp. 3370–3379.
[43] I. Amara, M. Ziaeefard, B. H. Meyer, W. Gross, and J. J. Clark, "Ces-kd: Curriculum-based expert selection for guided knowledge distillation," in *Proc. 26th International Conference on Pattern Recognition (ICPR)*, Montreal, QC, Canada, 2022, pp. 1901–1907.
[44] Z. Li, X. Li, L. Yang, B. Zhao, R. Song, L. Luo *et al.*, "Curriculum temperature for knowledge distillation," in *Proc. 37th Annual AAAI Conference on Artificial Intelligence (AAAI)*, Washington, DC, USA, 2023, pp. 1504–1512.
[45] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 29, pp. 3451–3460, Oct. 2021.
[46] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. 33th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 9726–9735.
[47] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. 20th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, vol. abs/1904.08779, pp. 2613–2617, Dec. 2019.
[48] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, May. 1992.
[49] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, pp. 1057–1063, 2000.
[50] H. Zou, Y. Si, C. Chen, D. Rajan, and E. S. Chng, "Speech emotion recognition with co-attention based multi-level acoustic information," in *Proc. 45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Singapore, 2022, pp. 7367–7371.
[51] J. Ye, X.-C. Wen, Y. Wei, Y. Xu, K. Liu, and H. Shan, "Temporal modeling matters: A novel temporal emotional modeling approach for speech emotion recognition," in *Proc. 46th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5.
[52] D. Ong, S. Sun, J. Su, and B. Chen, "Mitigating linguistic artifacts in emotion recognition for conversations from tv scripts to daily conversations," in *Proc. 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Torino,Italia, 2024, pp. 11 319–11 324.
[53] S. Chen, X. Xing, W. Zhang, W. Chen, and X. Xu, "Dwformer: Dynamic window transformer for speech emotion recognition," in *Proc. 46th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5.
[54] S. Shen, F. Liu, and A. Zhou, "Mingling or misalignment? temporal shift for speech emotion recognition with pre-trained representations," in *Proc. 46th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5.
[55] W. Chen, X. Xing, X. Xu, J. Pang, and L. Du, "Speechformer++: A hierarchical efficient framework for paralinguistic speech processing," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, no. 1, pp. 775–788, Feb. 2023.
[56] D. Hu, X. Hou, L. Wei, L. Jiang, and Y. Mo, "Mm-dfn: Multimodal dynamic fusion network for emotion recognition in conversations," in *Proc. 45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Singapore, 2022, pp. 7037–7041.

Ziping Zhao is a full professor of Computer Science at Tianjin Normal University. He received his Ph.D. for his study on the automatic prediction of prosodic phrases in 2008 from Nankai University, China. In 2018, he studied in the Chair of Embedded Intelligence for Health Care and Wellbeing at the University of Augsburg, Germany as a visiting scholar. In 2016, he became the vice dean of the college of computer and information engineering at Tianjin Normal University. He has published more than 30 publications in peer reviewed books, journals, and conference proceedings, including ICASSP, INTERSPEECH, Neural Networks and IEEE JSTSP. His research fields are affective computing and machine learning.

Jixin Liu is a post-graduate student in the College of Computer and Information Engineering, Tianjin Normal University. Her research interests include speech emotion recognition and applications.

Haishuai Wang is currently a ZJU100 Professor in the Department of Computer Science at Zhejiang University. Prior to that, he was a faculty member at Fairfield University and Harvard University. He received PhD of Computer Science from University of Technology Sydney, and did postdoc training at Washington University in St Louis and Harvard University. His research focuses on data mining and health informatics.

Danushka Bandara received the bachelor's degree in Electrical Engineering from the University of Moratuwa, Sri Lanka, in 2009. He received his master's and Ph.D. degrees in Computer Engineering and Electrical and Computer Engineering from Syracuse University, Syracuse, NY, USA, in 2013 and 2018, respectively. From 2019 to 2020, he worked as a Data Scientist at Corning Incorporated, Corning, NY, USA. Currently, he is an Assistant Professor of Computer Science and Engineering at Fairfield University, Fairfield, CT, USA. His Current research interests include Applied machine learning, Bioinformatics, Human-computer interaction, and Computational social science.

Jianhua Tao received the Ph.D. degree from Tsinghua University, Beijing, China, in 2001, and the M.S. degree from Nanjing University, Nanjing, China, in 1996. He is currently a Professor with NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include speech synthesis and coding methods, human computer interaction, multimedia information processing, and pattern recognition. He has authored or coauthored more than 80 papers on major journals and proceedings including IEEE Transactions on Audio, Speech, and Language Processing, and received several awards from the important conferences, such as Eurospeech, NCMMSC, etc. He serves as the chair or program committee member for several major conferences, including ICPR, ACII, ICMI, ISCSLP, NCMMSC, etc. He also serves as the steering committee member for IEEE Transactions on Affective Computing, an Associate Editor for Journal on Multimodal User Interface and International Journal on Synthetic Emotions, the Deputy Editor-in-Chief for Chinese Journal of Phonetics.