Relationship-Guided Knowledge Transfer for Class-Incremental Facial Expression Recognition

Yuanling Lv, Yan Yan, *Senior Member, IEEE,* Jing-Hao Xue, *Senior Member, IEEE,*
Si Chen, *Member, IEEE,* and Hanzi Wang, *Senior Member, IEEE*

***Abstract*—Human emotions contain both basic and compound facial expressions. In many practical scenarios, it is difficult to access all the compound expression categories at one time. In this paper, we investigate comprehensive facial expression recognition (FER) in the class-incremental learning paradigm, where we define well-studied and easily-accessible basic expressions as initial classes and learn new compound expressions incrementally. To alleviate the stability-plasticity dilemma in our incremental task, we propose a novel Relationship-Guided Knowledge Transfer (RGKT) method for class-incremental FER. Specifically, we develop a multi-region feature learning (MFL) module to extract fine-grained features for capturing subtle differences in expressions. Based on the MFL module, we further design a basic expression-oriented knowledge transfer (BET) module and a compound expression-oriented knowledge transfer (CET) module, by effectively exploiting the relationship across expressions. The BET module initializes the new compound expression classifiers based on *expression relevance* between basic and compound expressions, improving the plasticity of our model to learn new classes. The CET module transfers *expression-generic knowledge* learned from new compound expressions to enrich the feature set of old expressions, facilitating the stability of our model against forgetting old classes. Extensive experiments on three facial expression databases show that our method achieves superior performance in comparison with several state-of-the-art methods.**

***Index Terms*—Facial expression recognition, Class-incremental learning, Knowledge transfer.**

I. Introduction

FACIAL expression is one of the most natural non-verbal signals to convey human emotions and intentions. It plays a vital role in our daily communications and social interactions. Over the past few decades, automatic facial expression recognition (FER) has attracted considerable attention due to its practical importance in a wide range of applications, including human-computer interaction, driver monitoring, etc. Benefiting from the recent progress of deep neural networks (DNN), a variety of FER methods [1]–[5] have been developed and achieved excellent performance under pose variations, occlusion, and noisy labels. However, these methods generally work only on basic expressions (including angry, disgusted, fearful, happy, sad, surprised, and neutral) according to Ekman and Friesen's study [6].

Y. Lv, Y. Yan, and H. Wang are with the Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, and the Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, Xiamen 361005, China (e-mail: lvyuanling@stu.xmu.edu.cn; yanyan@xmu.edu.cn; hanzi.wang@xmu.edu.cn).

J.-H. Xue is with the Department of Statistical Science, University College London, London WC1E 6BT, UK (e-mail: jinghao.xue@ucl.ac.uk).

S. Chen is with the School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China (e-mail: chensi@xmut.edu.cn).

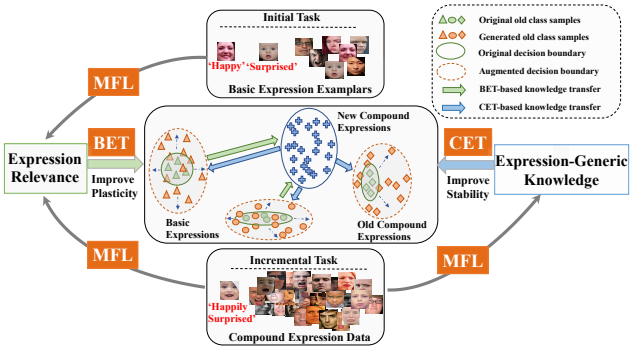

Fig. 1. **An illustration of our RGKT method.** The MFL module extracts fine-grained features for capturing subtle differences in expressions. The BET and CET modules exploit *expression relevance* and *expression-generic knowledge* to improve the model plasticity and stability, respectively.

Recently, Du *et al.* [7] reveal that basic expressions cannot comprehensively cover the diversity of human emotions and they define 22 expression categories, which can be classified into two groups (i.e., basic expressions and compound expressions). Compared with basic expressions, compound expressions are more fine-grained and more difficult to be annotated [8]. Later, the EmotioNet [9] and RAF-DB datasets [2] are laboriously collected to involve compound expressions in the wild under the professional instruction of psychology. A few DNN-based methods [10], [11] have been proposed to identify compound expressions.

Regrettably, in many practical scenarios, it is usually challenging to collect, annotate, and access the data containing various compound expression categories all at once, due to their diversity as well as some temporary constraints on privacy and device. Therefore, it is of great significance for an FER model to be able to continuously adapt to the ever-changing environment. Class-incremental learning (CIL), which aims to learn new classes incrementally without forgetting old classes, has emerged as a prominent learning paradigm. In this paper, we study a new and practical setup for class-incremental FER, where we define well-studied and easily-accessible basic expressions as initial classes while learning new compound expressions incrementally. Such a way greatly alleviates the requirement for data of all compound expression categories available at one time.

Due to memory limitations or data privacy, conventional

DNN models tend to adjust the network to new tasks but forget the previously learned knowledge (known as catastrophic forgetting [12]). Hence, many efforts on CIL [13]–[19] have been made to prevent catastrophic forgetting. They either impose constraints on old classes or introduce extra modules to accommodate new classes. However, these methods still suffer from the stability-plasticity dilemma [20] which refers to the trade-off between adapting to new concepts and preserving the old knowledge. This dilemma is caused by serious data imbalance between old and new classes (e.g., only a tiny set of exemplars from old classes are allowed to be used during incremental learning when the rehearsal strategy is used).

Different from the popular CIL tasks (such as general natural image classification), our FER task involves both basic and compound expressions, which exhibit great relevance at the semantic level. As shown in Fig. 1, the 'Happy', 'Surprised', and 'Happily Surprised' expressions are closely related. Moreover, different expressions share the generic knowledge (intra-class intensity variations). Hence, we can leverage such intimate relationship across expressions to perform expression recognition in the CIL paradigm.

To this end, we propose a Relationship-Guided Knowledge Transfer (RGKT) method for class-incremental FER. The architecture of RGKT consists of three main modules, including a multi-region feature learning (MFL) module, a basic expression-oriented knowledge transfer (BET) module, and a compound expression-oriented knowledge transfer (CET) module. An illustration of our method is given in Fig. 1. Specifically, on the one hand, we design an MFL module to learn fine-grained features for capturing subtle differences in expressions. This is highly beneficial for knowledge transfer across expressions. On the other hand, we develop the BET and CET modules by exploiting *expression relevance* and *expression-generic knowledge*, respectively. The BET module initializes the new compound expression classifiers considering the relevance between basic and compound expressions. The CET module transfers the expression-generic knowledge of new compound expressions to enlarge the feature set of old expressions. The design of the BET and CET modules greatly alleviates data imbalance between old and new classes.

The main contributions of this paper are as follows:

- We propose a novel RGKT method for class-incremental FER. By effectively transferring relationship-guided knowledge across expressions based on fine-grained features, we can incrementally identify new compound expressions without forgetting old expressions. To the best of our knowledge, we are the first to classify both basic and compound expressions in the CIL paradigm.
- We develop the BET and CET modules for powerful knowledge transfer. In particular, the CET module leverages curriculum learning to learn expression-generic knowledge and perform enhanced classification at the early and later stages of incremental learning, respectively. By tightly combining and jointly training these modules end-to-end, we can largely mitigate the stability-plasticity dilemma in our incremental task.
- Without bells and whistles, our method strikes a better balance between old and new classes than several state-of-the-art methods on facial expression databases.

The remainder of this paper is organized as follows. First, we review the related work in Section II. Then, we elaborately describe our proposed RGKT method in Section III. Next, we perform experiments on three facial expression databases in Section IV. Finally, we draw the conclusion in Section V.

II. Related Work

In this section, we first introduce facial expression recognition in Section II-A. Then, we briefly review class-incremental learning in Section II-B.

A. Facial Expression Recognition (FER)

Recent FER methods [2], [3], [5], [21], [22] either reduce the influence of various disturbing factors (such as pose and identity) or noisy labels in facial images, or extract discriminative expression features. These methods mainly target the classification of basic expressions (i.e., basic FER).

Du *et al.* [7] verify the diversity of expressions and define 22 expression categories (including basic expressions and compound expressions), where compound expressions are constructed by combinations of basic expressions. Note that compound expression images are not simply the addition of basic expression images. Instead, they involve complex facial motions that differ from those in basic expression images. These compound expressions are more fine-grained and show subtle differences, which can make the FER task more challenging. Subsequently, a large-scale EmotioNet dataset [9] and a real-world RAF-DB dataset [2] are collected with compound expression data. Slimani *et al.* [11] propose a highway network to classify compound expressions. Li *et al.* [10] learn appearance and geometric representations for compound FER. Compared with basic FER, research on compound FER is still in its infancy.

Due to the ever-changing environment in real-world applications, annotating and accessing all compound expression categories at one time can be struggling. Unlike existing methods, we investigate comprehensive FER for both basic and compound expressions in the CIL paradigm.

B. Class-Incremental Learning (CIL)

Existing CIL methods can be roughly divided into regularization-based methods [23]–[25], distillation-based methods [14], [15], [26], [27], and structure-based methods [16], [18], [28]. Regularization-based methods first estimate the importance of network parameters, and then prevent the parameters important to the old model from large changes. Distillation-based methods explicitly enforce the outputs of the new model to be similar to those of the old model. iCaRL [15] computes the distillation loss by using old exemplars and PODNet [14] proposes a spatial-based distillation loss. AFC [26] restricts the update of important features via the distillation loss. Structure-based methods separately learn the network parameters at the different stages to avoid undesirable overlapping between old and new classes. DER [18] develops

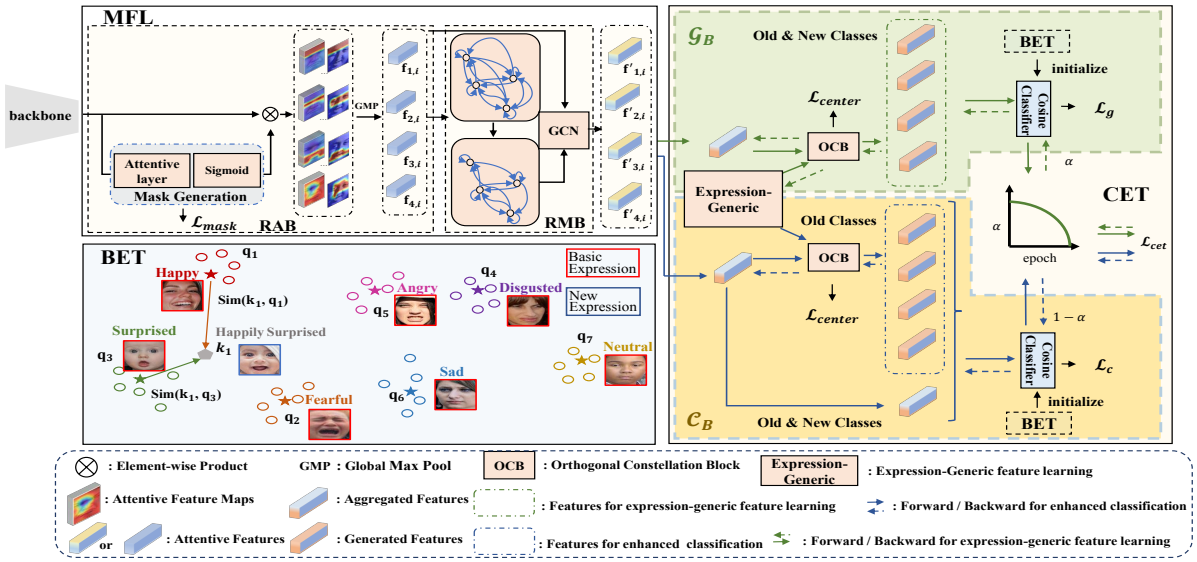

Fig. 2. **Overview of our proposed RGKT method.** RGKT consists of a backbone and three modules (an MFL module, a BET module, and a CET module). The MFL module contains RAB and RMB. The CET module contains an expression-generic feature learning branch $\mathcal{G}_B$ and an enhanced classification branch $\mathcal{C}_B$. The features in the CET module contain two parts: the original features (i.e., the aggregated features from MFL) and the generated features. The original features for old and new classes are used to learn expression-generic knowledge in $\mathcal{G}_B$. Based on this, the new features for old classes are generated in $\mathcal{C}_B$. The difference between the two branches is that only the gradient from $\mathcal{G}_B$ is used to update the expression-generic feature while $\mathcal{C}_B$ takes advantage of the expression-generic feature to enrich the features of old classes.

an effective channel-level mask-based pruning method. FOSTER [16] proposes a feature compression method to remove the model redundancy.

The above CIL methods are usually designed for the general natural image classification task. Unlike this task, our FER task focuses on classifying various expressions, among which the class-generic information is shared. Thus, we propose to exploit the expression-generic knowledge learned from new classes (with a relatively large amount of training data) and transfer it to old classes, reducing catastrophic forgetting.

Recently, Zhu *et al.* [29] first investigate the FER task in the CIL setting. They develop a center-expression-distilled loss to extract features. However, they aim at basic FER and do not explore the relationship between basic and compound expressions. Thus, such a method is difficult to generalize to identify hardly collected compound expressions. Different from this method, we study CIL in a more practical setting, which covers both basic and compound expressions. In particular, we select well-studied and easily-accessible basic expressions as initial classes and incrementally learn new compound expressions.

III. Methodology

In this section, we introduce our RGKT method for class-incremental FER. First, we give the problem formulation in Section III-A. Then, we provide the overview of RGKT in Section III-B. Next, we introduce the key components (including the MFL module, the BET module, and the CET module) of RGKT in Sections III-C, III-D, and III-E, respectively. Subsequently, we give the joint loss in Section III-F. Finally, we summarize the overall training in Section III-G.

A. Problem Formulation

To establish a practical setting for class-incremental FER, we define the classification of well-studied and easily-accessible basic expressions as the initial task and identify new compound expressions incrementally. Assume that we have a sequence of N+1 training tasks, containing an initial task and N incremental tasks. Accordingly, the training data are represented as $\{\mathcal{D}^0, D^1, \cdots, \mathcal{D}^N\}$, where $\mathcal{D}^0$ and $\mathcal{D}^n$ are the initial training subset and the n-th incremental subset, respectively. Note that the expression categories from different subsets are disjoint.

Following the rehearsal strategy [15], we store a tiny subset of exemplars from old classes as the memory and fix these exemplars at each incremental task. For the n-th incremental task, we use limited exemplars (denoted $\mathcal{E}^n$) from old classes and all the samples $\mathcal{D}^n$ from new classes to constitute the training set. At each iteration of the training phase, one mini-batch $\mathcal{B}_n = \{(\mathbf{x}_i^n, y_i^n)\}_{i=1}^b$ is randomly sampled from the training set, where $\mathbf{x}_i^n \in \{\mathcal{E}^n \cup \mathcal{D}^n\}$ and $y_i^n \in \mathcal{Y}^n$ denote the i-th input image and its corresponding label, respectively; $\mathcal{Y}^n$ is the label set of old and new classes; b is the batch size. During the testing phase, we evaluate the trained model on the test data including both old and new classes seen so far.

B. Overview

An overview of our RGKT method is illustrated in Fig. 2. Given a batch of facial images from an incremental subset and a few exemplars from old classes, we first feed them into a backbone to extract preliminary feature maps. To capture subtle differences in compound expressions and promote knowledge transfer at the incremental task, we develop an

MFL module including a region attention block (RAB) and a relational modeling block (RMB) to further extract fine-grained features based on preliminary feature maps. RAB exploits high-level semantic information and obtains multiple attentive features. RMB updates these features by uncovering the interconnections between them.

Based on the MFL module, we design two knowledge transfer modules to alleviate the stability-plasticity dilemma by effectively modeling the underlying relationship between facial expressions. On the one hand, we introduce a BET module to initialize the new class classifiers based on expression relevance between basic and compound expressions. Hence, our method can adapt to new classes quickly, enhancing the model plasticity. On the other hand, we propose a CET module to transfer the learnable expression-generic knowledge from new classes with a relatively large amount of samples to old classes (for which only a tiny set of exemplars are stored in memory), generating new features of old classes. Such a way enriches the feature set of old classes, mitigating the forgetting of old classes and thus improving the model stability.

C. Multi-Region Feature Learning (MFL) Module

Region Attention Block (RAB). To encode fine-grained appearance variations, we design RAB to extract multiple attentive features from different facial regions with attention masks. Technically, given the i-th input facial image $\mathbf{x}_i^n$ at the n-th incremental task, the feature map extracted by the backbone is denoted as $\mathbf{P}_i \in \mathbb{R}^{H' \times W' \times C'}$, where H', W', and C' are the height, width, and channel number, respectively. Based on $\mathbf{P}_i$, we first generate T 2-dimensional masks as

$$\mathbf{M}_{t,i} = \mathrm{V}_t(\mathbf{P}_i), t = 1, \cdots, T, \tag{1}$$

where $\mathrm{V}_t(\cdot)$ denotes the mask generation operation (consisting of a convolutional layer followed by a Sigmoid function) and $\mathbf{M}_{t,i} \in \mathbb{R}^{H' \times W'}$ represents the generated mask.

Then, we obtain T attentive convolutional feature maps

$$\mathbf{P}_{t,i}^{'} = \mathrm{S}(\mathbf{M}_{t,i}) \otimes \mathbf{P}_i, t = 1, \cdots, T, \tag{2}$$

where $\mathrm{S}(\cdot)$ is a reshape function which resizes the input to the same size as $\mathbf{P}_i$ and $\otimes$ is the element-wise product.

Finally, T attentive feature maps are fed into parallel global max pooling (GMP) layers, and thus T attentive features $\{\mathbf{f}_{1,i}, \mathbf{f}_{2,i}, \cdots, \mathbf{f}_{T,i}\}$, where $\mathbf{f}_{t,i} \in \mathbb{R}^D$ and D represents the dimension of each feature, are extracted for $\mathbf{x}_i^n$.

To ensure that each mask can target a specific facial region (e.g., when T=4, the first 3 masks focus on upper (eyes and eyebrows), middle (nose), and lower facial regions (mouth), while the last mask emphasizes the whole facial region), we vertically divide each of the first T-1 masks into T-1 uniform patches. In this paper, the division is done manually since we expect the model can focus on meaningful facial regions. Hence, we define the mask loss for the i-th image as

$$\mathcal{L}_{mask} = \sum_{t=1}^{T-1} \Big(\sum_{(x,y) \in \bar{\mathbf{M}}_{t,i}} \bar{\mathbf{M}}_{t,i}(x,y) - \sum_{(x,y) \in \mathbf{R}_{t,t}} \bar{\mathbf{M}}_{t,i}(x,y) \Big), \tag{3}$$

where $\bar{\mathbf{M}}_{t,i}(x,y) = \mathrm{Sigmoid}(\mathbf{M}_{t,i}(x,y)) - I_{t,i}$ and $I_{t,i} = 1/(h \times w) \sum_{(x,y) \in \mathbf{M}_{t,i}} \mathbf{M}_{t,i}(x,y)$ indicate the t-th normalized mask and the average value of $\mathbf{M}_{t,i}$, respectively; h and w are the height and width of the mask, respectively; $\mathrm{Sigmoid}(\cdot)$ is the sigmoid operation to normalize $\mathbf{M}_{t,i}$; $\mathbf{R}_{t,t}$ represents the t-th patch of the t-th normalized mask. Minimizing the mask loss (which subtracts the values in $\mathbf{R}_{t,t}$ from the t-th mask) enforces the responses of the mask out of $\mathbf{R}_{t,t}$ to be reduced and thus highlights $\mathbf{R}_{t,t}$ (some generated masks are given in Fig. 2).

Note that Ruan *et al.* [3] propose to extract a set of latent features, which are learned by an unsupervised compactness loss. In RAB, we explicitly associate each attentive feature with a specific facial region by a mask loss, beneficial for subsequent relational modeling and compound FER.

Relational Modeling Block (RMB). In RAB, multiple attentive features are individually extracted for each facial image. In other words, RAB does not consider the connections between these features. In fact, each expression usually involves various action units, which correspond to different facial regions. Hence, we leverage RMB to explore the interconnections between attentive features.

Specifically, we model the interconnections between features as a graph, where each attentive feature is viewed as a vertex. The similarity between two vertices $\mathbf{f}_{m,i}$ and $\mathbf{f}_{n,i}$ is given as $\mathrm{Sim}(\mathbf{f}_{m,i}, \mathbf{f}_{n,i}) = \mathbf{f}_{m,i}^{\mathrm{T}} \mathbf{f}_{n,i}$. Next, a K-nearest neighbor (KNN) graph can be constructed, where two vertices are connected by an edge, if the similarity between them is among the K largest similarities of one vertex (K=2 in this paper). The KNN graph is further fed into a graph convolutional network (GCN) layer [30] to update the vertices. Finally, we combine updated features into an aggregated feature, defined as $\mathbf{f}_i = \sum_{t=1}^{T} \mathbf{f}_{t,i}^{'}$ ($\mathbf{f}_{t,i}^{'} \in \mathbb{R}^D$ is the t-th updated feature for the i-th image), which denotes the original expression feature.

D. Basic Expression-Oriented Knowledge Transfer (BET) Module

Compound expressions can be viewed as a meaningful combination of basic expressions [7]. For instance, the 'Happily Surprised' expression can be considered as a combination of 'Happy' and 'Surprised' expressions. Motivated by this, instead of randomly initializing the classifier weights for new classes (i.e., compound expressions), we can initialize them by leveraging the well-trained weights obtained for basic expressions according to expression relevance. In this way, the intrinsic correlation between basic and compound expressions is effectively incorporated into model training. As a result, our model can obtain good initialization parameters and thus quickly adapt to new classes.

Specifically, we calculate the prototypes (each prototype refers to the mean of one category) for a newly coming compound expression and its relevant basic expressions. Then, we obtain the similarities between compound and basic expressions via $\mathrm{Sim}(\cdot, \cdot)$. Thus, we initialize the classifier weights for a new compound expression as

$$\mathbf{W}_p^{new} = \sum_{l=1}^{L} (\mathrm{Sim}(\mathbf{k}_p, \mathbf{q}_l) \mathbf{W}_l^{basic}), \tag{4}$$

where $\mathbf{W}_p^{new}$ and $\mathbf{W}_l^{basic}$ denote the weights of the cosine classifiers for the p-th new compound expression and the l-th basic expression, respectively; $\mathrm{Sim}(\mathbf{k}_p, \mathbf{q}_l)$ denotes the similarity between the prototype $\mathbf{k}_p$ (corresponding to the p-th compound expression at the incremental task) and the prototype $\mathbf{q}_l$ (corresponding to the l-th basic expression at the initial task); L is the number of relevant basic expressions (L=2) w.r.t. the p-th new compound expression.

E. Compound Expression-Oriented Knowledge Transfer (CET) Module

Due to a strict memory budget, only a small set of exemplars from old classes can be stored in memory, leading to extreme data imbalance between old and new classes. Although the knowledge distillation technique [31] is often leveraged to learn old classes using very limited exemplars, the trained model inevitably tends to have a strong bias towards exemplars without well preserving the whole knowledge of old classes. In our FER task, considering the high similarities between expressions, where different expressions share the generic knowledge (intra-class intensity variations) caused by similar facial motions. That is, the variations in different facial regions (i.e., AUs in FACS) are similar for these expressions. Therefore, we can enrich the feature set of old classes by transferring the expression-generic knowledge (shared among all the expressions) learned from new classes (having a relatively large amount of training data) to old classes, avoiding relying only on the distillation of limited knowledge from the exemplars of old classes.

Based on the above observations, we develop a CET module, which contains an expression-generic feature learning branch $\mathcal{G}_B$ and an enhanced classification branch $\mathcal{C}_B$, to effectively learn expression-generic features from new classes and use these features to augment the feature set of old classes. The detailed architecture of the CET module is given in Fig. 2.

Both branches consist of an expression-generic feature learning block (including a set of learnable expression-generic features), an orthogonal constellation block (generating new expression features), and a cosine classifier [32], where the expression-generic feature learning block is shared by two branches. For $\mathcal{G}_B$, we expect that the expression-generic features can be learned to capture rich intra-class variations from new classes since new classes are dominant for learning the generic knowledge. This makes the classifier tend to fit a large number of new class samples, and thus facilitates the transfer of compound expression-oriented knowledge to old expressions. Meanwhile, for $\mathcal{C}_B$, we enrich the feature set of old classes based on the expression-generic features learned from new classes. This can alleviate the influence of data imbalance between old and new classes. As a result, old class classifiers are able to learn more diversity to offset the impact of small memory. During the testing phase, only $\mathcal{C}_B$ is used to classify both old and new classes.

Orthogonal Constellation Block (OCB). Recently, Yang *et al.* [33] jointly train local and global features in an orthogonal fusion to perform image retrieval. Inspired by this, we develop OCB to remove the components (that are overlapped with expression-generic features) from the original expression feature and extract expression-specific features. Both the two branches share the same OCB. In this way, we can generate a set of compact features.

Mathematically, we first calculate a set of projections $\{\mathbf{p}_{1,i}, \cdots, \mathbf{p}_{M,i}\}$ based on the original expression feature $\mathbf{f}_i$ and a set of learnable expression-generic features $\{\mathbf{f}_{1,generic}, \cdots, \mathbf{f}_{M,generic}\}$, that is,

$$\mathbf{p}_{m,i} = \frac{\mathbf{f}_i \cdot \mathbf{f}_{m,generic}}{||\mathbf{f}_{m,generic}||_2^2}\mathbf{f}_{m,generic}, m = 1, \cdots, M, \quad (5)$$

where $\mathbf{f}_i \cdot \mathbf{f}_{m,generic}$ represents the dot product operation and $||\mathbf{f}_{m,generic}||_2^2$ is the squared L_2 norm of $\mathbf{f}_{m,generic}$; M is the number of learned features. These projections indicate the components that are overlapped with expression-generic features in the original expression feature.

Then, we can obtain the orthogonal components of the original expression feature, which represent the expression-specific features for one expression category. That is,

$$\mathbf{o}_{m,i} = \mathbf{f}_i - \mathbf{p}_{m,i}, m = 1, \cdots, M. \quad (6)$$

To ensure that the orthogonal components accurately encode the class-specific information, we leverage a center loss to penalize the distances between orthogonal components and their corresponding prototype for the i-th image

$$\mathcal{L}_{center} = \sum_{m=1}^{M} ||\mathbf{o}_{m,i} - \mathbf{c}||_2^2, \quad (7)$$

where $\mathbf{c}$ denotes the prototype of the expression category that $\mathbf{x}_i^n$ belongs to.

Finally, we concatenate the expression-generic features and the expression-specific features, and obtain a set of generated expression features for old and new classes.

Classification Losses. Given a mini-batch $\mathcal{B}_n$, we generate a number of expression features. Suppose that $\mathbf{f}_g$ comes from the generated expression features for old and new classes in $\mathcal{G}_B$, while $\mathbf{f}_c$ comes from the generated expression features for old classes and the original expression features for both old and new classes in $\mathcal{C}_B$. The classification losses of two branches can be formulated as

$$\mathcal{L}_g = -\sum_{j \in \mathcal{Y}_n} \mathbb{1}_{[j=y_g]} \log(\theta_g(\mathbf{f}_g)), \quad (8)$$

$$\mathcal{L}_c = -\sum_{j \in \mathcal{Y}_n} \mathbb{1}_{[j=y_c]} \log(\theta_c(\mathbf{f}_c)), \quad (9)$$

where $\mathcal{L}_g$ and $\mathcal{L}_c$ are the losses for the expression-generic feature learning branch and the enhanced classification branch, respectively; θ_g and θ_c are two cosine classifiers; y_g and y_c are the expression labels which $\mathbf{f}_g$ and $\mathbf{f}_c$ respectively belong to. When $j = y_g$ or $j = y_c$, the function $\mathbb{1}_{[j=y_g]}$ or $\mathbb{1}_{[j=y_c]}$ is equal to 1; otherwise its value is 0.

Note that only the gradient from $\mathcal{L}_g$ is used to learn expression-generic features during back-propagation. Such a manner ensures that expression-generic features are mainly learned from new classes, whose training samples are much more than the exemplars of old classes.

Curriculum Learning. To fully exploit the unique roles of the two branches, we expect that at the early learning stage, the network can focus on capturing the expression-generic knowledge from new classes to enrich the feature set of old classes. Meanwhile, at the later learning stage, the network can gradually excel in the classification of old and new classes. To achieve this, we leverage curriculum learning to balance the two branches by an adaptive weight. In this way, the two branches are trained to benefit each other to boost the final performance.

The loss of the CET module is defined as

$$\mathcal{L}_{cet} = \alpha \mathcal{L}_g + (1-\alpha)\mathcal{L}_c. \tag{10}$$

Here the adaptive weight α is calculated by $1-(T/T_{max})^{\beta}$, (we set $\beta = 2$ as [34]), where T and T_{max} are the current epoch and the total number of epochs in the training phase, respectively; that is, the value of α gradually decreases with the increasing epochs.

F. Joint Loss

Based on the above formulation, the joint loss is given as

$$\mathcal{L}_{joint} = \lambda_1 \mathcal{L}_{mask} + \lambda_2 \mathcal{L}_{center} + \lambda_3 \mathcal{L}_{cet}, \tag{11}$$

where λ_1, λ_2, and λ_3 are the balance weights.

G. Overall Training

We summarize the overall training of our method at the n-th incremental task in Algorithm 1.

Algorithm 1 The overall training of our method at the n-th incremental task

Input: The n-th incremental subset $\mathcal{D}^n$; the old class exemplars $\mathcal{E}^n$; the total training epochs T_{max};
Output: The updated model M;

Initialize the new compound expression classifiers via Eq. (4) based on the BET module;
for each t = 1 to T_{max} **do**
  for each mini-batch in $\{\mathcal{D}^n \cup \mathcal{E}^n\}$ **do**
    Obtain the orignal expression features $\{\mathbf{f}^i_{new} \cup \mathbf{f}^j_{old}\}$ of new and old classes based on the MFL module and calculate the mask loss via Eq. (3);
    Obtain the generated expression features $\{\mathbf{f'}^i_{new} \cup \mathbf{f'}^j_{old}\}$ of new and old classes based on the CET module and calculate the center loss via Eq. (7);
    Calculate the adaptive weight $\alpha \leftarrow 1-(T/T_{max})^2$;
    for each feature from $\{\mathbf{f'}^i_{new} \cup \mathbf{f'}^j_{old}\}$ **do**
      Calculate the classification loss of $\mathcal{G}_B$ via Eq. (8);
    end for
    for each feature from $\{\mathbf{f}^i_{new} \cup \mathbf{f}^j_{old} \cup \mathbf{f'}^j_{old}\}$ **do**
      Calculate the classification loss of $\mathcal{C}_B$ via Eq. (9);
    end for
    Calculate the joint loss via Eq. (11) ;
    Update the model M by stochastic gradient descent (SGD);
  end for
end for

IV. Experiments

In this section, we first introduce the databases in Section IV-A. Then, we present the implementation details of our method in Section IV-B. Next, we conduct ablation studies in Section IV-C and give some visualization results in Section IV-D. Finally, we compare our method with state-of-the-art methods in Section IV-E.

A. Databases

RAF-DB [2] contains diverse real-world images collected from the Internet, which are manually crowd-sourced annotated and reliable estimation. It includes seven basic expressions with 15,339 images (12,271 training images and 3,068 test images) and eleven compound expressions with 3,954 images (3,162 training images and 792 test images). **CFEE** [7] is collected from 230 human subjects and it contains seven basic expressions (with 1,610 images) and fifteen compound expressions (with 3,450 images). **EmotioNet** [9] is a large-scale in-the-wild database collected from the Internet. We use the second track of the EmotioNet Challenge. It provides 2,478 images with six basic expressions and ten compound expressions. Note that some popular facial expression databases (such as AffectNet [35]) cannot be used for evaluation since they involve only basic expressions.

B. Implementation Details

For all the experiments, we first align and crop facial images to the size of 256×256, and then resize them to 224×224. All the results are obtained based on PyCIL [36] (a python toolbox for CIL) under the same settings. We adopt ResNet-18 [37] as the backbone, and train the whole network using stochastic gradient descent (SGD) [38] with an initial learning rate of 0.01 at the initial task and 0.001 at the incremental task, where we use CosineAnnealingLR [39] as a scheduler. All the tasks are trained for 40 epochs with a batch size of 32. The number of exemplars for each old class is set to 20.

Following the common settings of CIL in [15], we first train our method on basic expression data as the initial task. Then, we fix the number of incremental classes (compound expressions) to C=3 or C=5 at the incremental tasks. The number of attentive features T is set to 4. The number of expression-generic features M is set to 10. The balance weights λ_1, λ_2, and λ_3 are empirically set to 0.01, 0.01, and 1.00, respectively. During the testing phase, we evaluate our method on the classes (including both old and new expressions) ever seen so far. We report the average accuracy as well as the standard deviation, as done in [18].

C. Ablation Studies

The details of several variants of our RGKT method are shown in Table I. The results obtained by different variants of RGKT on RAF-DB are given in Table II. ResNet-18 with a cosine classifier is used as our baseline method.

TABLE I
THE DETAILS OF SOME VARIANTS OF OUR RGKT METHOD. 'OCB' AND 'CL' INDICATE 'ORTHOGONAL CONSTELLATION BLOCK' AND 'CURRICULUM LEARNING', RESPECTIVELY. † INDICATES THE BRANCH IS TRAINED WITH THE SAME NUMBER OF FEATURES FOR OLD AND NEW CLASSES IN EACH BATCH (BY MULTIPLE COPIES OF EXEMPLARS FOR OLD CLASSES).

Methods	Branch Type	OCB	CL
Baseline+MFL+CET (G)	Expression-Generic Feature Learning Branch	✘	✘
Baseline+MFL+CET (C)	Enhanced Classification Branch†	✘	✘
Baseline+MFL+CET (T)	Two Branches	✘	✘
Baseline+MFL+CET (TO)	Two Branches	✔	✘
Baseline+MFL+CET	Two Branches	✔	✔

Influence of the MFL Module. From Table II, compared with Baseline, Baseline+MFL (RAB), which extracts multiple attentive features, obtains much higher accuracy for C=3 and C=5. Hence, it is vital to extract fine-grained features by considering the influence of different facial regions. By jointly combining RAB and RMB in MFL, Baseline+MFL (RAB+RMB) achieves better results than Baseline+MFL (RAB). This shows the importance of RMB, which models the interconnections between attentive features via a graph. Moreover, by adding the MFL module into Baseline+CET, Baseline+MFL+CET achieves 2.11% and 3.41% improvements in terms of average accuracy for C=3 and C=5, respectively. These results validate that fine-grained features can greatly facilitate knowledge transfer during incremental learning.

Influence of the CET Module. From Table II, Baseline+MFL+CET (T) obtains much higher accuracy (6.41% and 5.86% improvements for C=3 and C=5, respectively) than Baseline+MFL+CET (G), showing that the two branches can benefit each other to boost the performance. Note that Baseline+MFL+CET (G) gets better performance than Baseline+MFL by transferring the expression-generic knowledge. However, Baseline+MFL+CET (G) only focuses on capturing intra-class variations while ignoring the classification of fine-grained expressions. Baseline+MFL+CET (C) simply increases the number of old classes by multiple copies of exemplars. Such a way does not fully exploit the knowledge from new classes. By applying OCB, Baseline+MFL+CET (TO) outperforms Baseline+MFL+CET (T). This is because OCB removes overlapped components in the original expression features and generates compact features. Baseline+MFL+CET gives higher accuracy than Baseline+MFL+CET (TO) since curriculum learning is employed to balance the trade-off between the two branches. The above results validate the necessity of each key component in the CET module.

We also evaluate the influence of the number of expression-generic features M in the CET module on the final performance. The results are given in Fig. 3. Our method obtains the best results when M is set to 10. When the value of M is too large, there exists redundant information between expression-generic features. Meanwhile, when the value of M is too small, the expression-generic features cannot be effectively learned from new classes.

Influence of the BET Module. Compared with Baseline+MFL+CET which randomly initializes the new class classifiers, Baseline+MFL+CET+BET achieves 1.51% and 1.78% accuracy improvements for C=3 and C=5, respectively. Thus, the BET module enables our model to obtain good initialization parameters, achieving better results.

TABLE II
ABLATION STUDIES FOR DIFFERENT VARIANTS OF OUR METHOD WITH THE NUMBERS OF INCREMENTAL CLASSES C=3 AND C=5 ON RAF-DB. 'AVG±STD' DENOTES THE AVERAGE ACCURACY (%) AND THE STANDARD DEVIATION OVER THE TRAINING TASKS. THE BEST RESULTS ARE MARKED IN **BOLD**.

Methods	Avg±std	
	C=3	C=5
Baseline	$44.12_{\pm 1.15}$	$45.13_{\pm 0.69}$
Baseline+MFL (RAB)	$50.95_{\pm 0.99}$	$50.86_{\pm 1.66}$
Baseline+MFL (RAB+RMB)	$51.41_{\pm 0.79}$	$51.30_{\pm 1.51}$
Baseline+CET	$66.72_{\pm 1.18}$	$66.72_{\pm 0.57}$
Baseline+MFL+CET (G)	$62.00_{\pm 1.40}$	$63.15_{\pm 1.40}$
Baseline+MFL+CET (C)	$63.50_{\pm 1.49}$	$63.70_{\pm 1.01}$
Baseline+MFL+CET (T)	$68.41_{\pm 1.52}$	$69.01_{\pm 0.93}$
Baseline+MFL+CET (TO)	$68.43_{\pm 0.72}$	$69.68_{\pm 1.07}$
Baseline+MFL+CET	$68.83_{\pm 0.90}$	$70.13_{\pm 0.96}$
Baseline+MFL+CET+BET	$\mathbf{70.34}_{\pm 0.96}$	$\mathbf{71.91}_{\pm 0.43}$

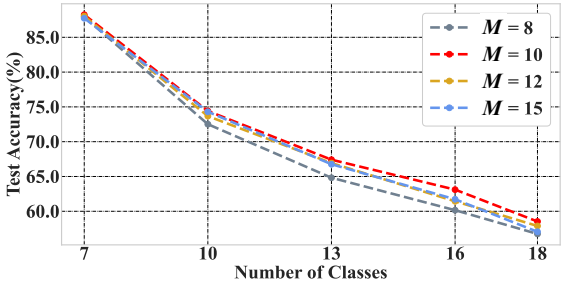

Fig. 3. **Ablation studies** for the number of expression-generic features M on RAF-DB. We report the test accuracy w.r.t. the number of classes during the whole incremental learning process.

Influence of the Number of Incremental Classes. We also conduct an ablation study to show the influence of the number of incremental classes (denoted as C) on RAF-DB. As shown in Fig. 4, we can see that our method significantly outperforms

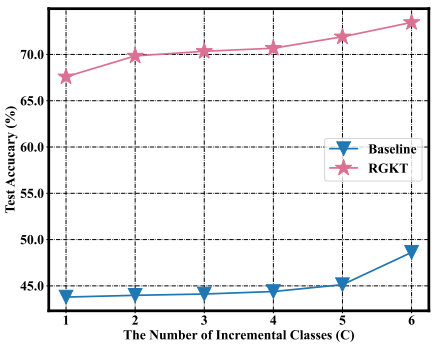

Fig. 4. **Ablation studies** for the influence of the number of incremental classes on RAF-DB.

TABLE III
ABLATION STUDIES FOR DIFFERENT VALUES OF β FOR THE ADAPTIVE WEIGHT CALCULATION α OF INCREMENTAL CLASSES C=3 AND C=5 ON RAF-DB. 'AVG±STD' DENOTES THE AVERAGE ACCURACY (%) AND THE STANDARD DEVIATION OVER THE TRAINING TASKS. THE BEST RESULTS ARE MARKED IN **BOLD**.

β	α	Avg±std	
		C=3	C=5
-	0.5	$68.43_{\pm 0.72}$	$69.68_{\pm 1.07}$
1	$1-(T/T_{max})$	$69.62_{\pm 1.22}$	$70.01_{\pm 0.56}$
1.5	$1-(T/T_{max})^{1.5}$	$69.77_{\pm 1.47}$	$70.14_{\pm 0.58}$
2	$1-(T/T_{max})^{2}$	$\mathbf{70.34}_{\pm 0.96}$	$\mathbf{71.91}_{\pm 0.43}$
2.5	$1-(T/T_{max})^{2.5}$	$69.83_{\pm 1.18}$	$70.14_{\pm 1.09}$

the Baseline method. The different values of C show different influences on the final performance. Both Baseline and our method get the worst performance when C=1 while getting the best performance when C=6. When the number of incremental classes becomes small, the performance drop of our model is more evident since the samples in new classes can be insufficient to learn shared generic knowledge. Note that in practical applications, it is usually easy to collect 2 or 3 classes for each incremental task. Therefore, considering the practicality of our setting, we set C=3 (when the incremental number is small) and C=5 (when the incremental number is large) to evaluate the performance of the CIL FER task.

Influence of the Adaptive Weight Calculation. We evaluate different values of β (β =1, β =1.5, β =2, and β =2.5) for the adaptive weight calculation. Table III shows the influence of different values of β, where the fixed weight ($\alpha = 0.5$) is also used for a comparison. We can see all the variants that adopt adaptive weights perform better than the method without using adaptive weights, since curriculum learning can encourage the model to pay different attention to two branches at different learning stages. For the adaptive weight calculation, when the value of β is 2, our method can achieve the best result.

Influence of the Number of Attentive Features. As shown in Fig. 5, we can see that our proposed method obtains the best performance when the number of attentive features (T) is set to 4. When the value of T is set to 4, such a division can effectively describe the meaningful components of the human face (i.e., upper, middle, and lower facial regions). On one hand, when a small number of attentive features are used, the key regions may not be located accurately and thus may ignore the important information. On the other hand, when a large number of attentive features are used, there exists redundancy among these features and the model cannot extract informative knowledge for learning. Therefore, we set the number of attentive features to 4.

Influence of the Parameters. We evaluate the performance of our method with the different values of λ_1, λ_2, and λ_3 in Eq. (11). The results are given in Table IV. Specifically, we first fix λ_2 = 0.01 and λ_3 = 1, and set the value of λ_1 from 0.001 to 1. Experimental results are shown in Table IV(a). We can observe that our method obtains the top performance when the value of λ_1 is set to 0.01. Then, Table IV(b) shows the results obtained by our method, when the values of λ_1 and λ_3 are set to 0.01 and 1, respectively, and the value of λ_2 varies from 0.001 to 1. When the value of λ_2 is set to 0.01, our method achieves the best results. Finally, we fix λ_1 = 0.01 and λ_2 = 0.01, and the range of λ_3 is from 0.01 to 1. As shown in Table IV(c), our method achieves outstanding performance when λ_3 =1. Therefore, our method obtains the best performance when the values of both λ_1 and λ_2 are set to 0.01 and the value of λ_3 is set to 1.

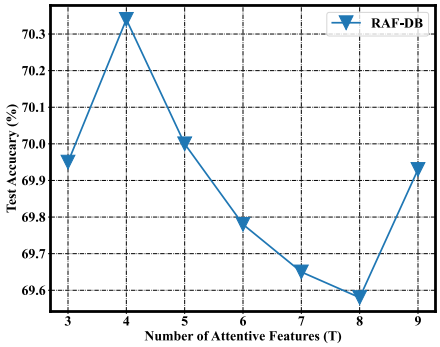

Fig. 5. **Ablation studies** for the different numbers of attentive features on the RAF-DB (C=3) databases.

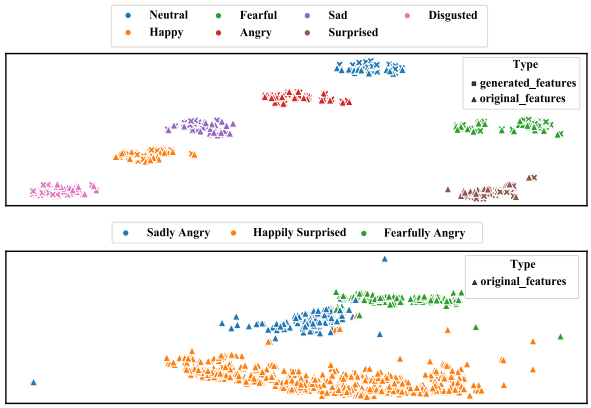

Fig. 6. **Feature visualization** of the original and generated features of old expressions (the upper panel) as well as the original features of new expressions (the lower panel) by using t-SNE [40] on RAF-DB.

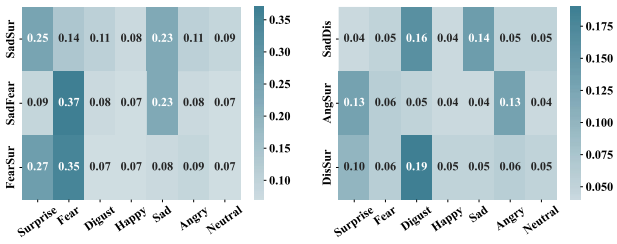

Fig. 7. **Visualization of similarities between expressions** on RAF-DB. The similarities between compound expressions and basic expressions at two different incremental tasks are computed. On each panel, the vertical axis represents the compound expressions at one incremental task while the horizontal axis represents the basic expressions.

TABLE IV
ABLATION STUDIES FOR THE DIFFERENT VALUES OF (A) λ_1, (B) λ_2, AND (C) λ_3 WITH THE DIFFERENT NUMBERS OF INCREMENTAL CLASSES C=3 AND C=5 ON RAF-DB. WE REPORT THE AVERAGE ACCURACY (%) AND THE STANDARD DEVIATION OVER THE TRAINING TASKS. THE BEST RESULTS ARE MARKED IN **BOLD**.

λ_1	C=3	C=5
0.001	$69.82_{\pm 1.44}$	$70.25_{\pm 1.48}$
0.01	$\mathbf{70.34}_{\pm 0.96}$	$\mathbf{71.91}_{\pm 0.43}$
0.1	$69.45_{\pm 1.33}$	$70.16_{\pm 1.41}$
1	$69.18_{\pm 1.29}$	$69.99_{\pm 1.81}$

(a) Influence of λ_1

λ_2	C=3	C=5
0.001	$69.10_{\pm 0.85}$	$69.85_{\pm 1.27}$
0.01	$\mathbf{70.34}_{\pm 0.96}$	$\mathbf{71.91}_{\pm 0.43}$
0.1	$69.47_{\pm 2.38}$	$69.73_{\pm 0.74}$
1	$54.40_{\pm 2.54}$	$55.20_{\pm 2.18}$

(b) Influence of λ_2

λ_3	C=3	C=5
0.01	$63.87_{\pm 3.84}$	$66.30_{\pm 1.29}$
0.1	$67.11_{\pm 0.93}$	$67.17_{\pm 1.52}$
1	$\mathbf{70.34}_{\pm 0.96}$	$\mathbf{71.91}_{\pm 0.43}$

(c) Influence of λ_3

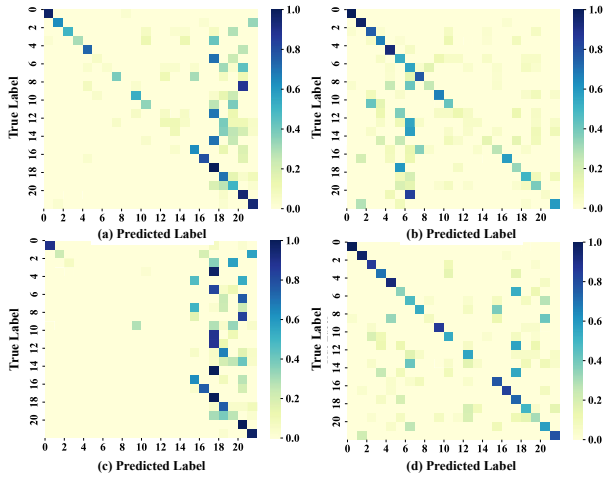

Fig. 8. **Visualization of confusion matrices** obtained by (a) Baseline, (b) FOSTER, (c) SCN, and (d) our RGKT method at the last incremental task on CFEE. The vertical axis represents the true label while the horizontal axis represents the predicted label.

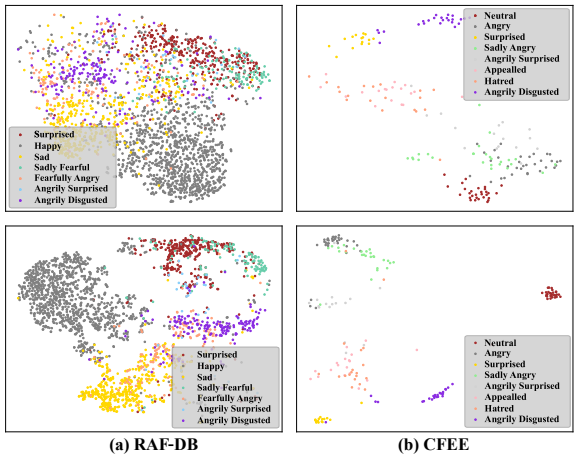

Fig. 9. **Visualization of expression features** on the (a) RAF-DB and (b) CFEE. We randomly select three basic expressions and compound expressions from different incremental tasks. The upper row and the lower row of panels show the feature distributions obtained by the Baseline method and our method, respectively.

D. Visualization

Visualization of Original and Generated Features. Fig. 6 gives feature visualization of the original and generated features of old expressions as well as the original features of new expressions at one incremental task. We can see that the generated features accurately encode the expression-generic knowledge learned from new classes and can greatly enrich the representations of old classes. Hence, we are able to significantly alleviate the forgetting of old classes.

Visualization of Similarities Between Expressions. In Fig. 7, we visualize the similarities between basic expressions and compound expressions on RAF-DB. The more similar the compound expressions and basic expressions are, the darker the colors are. In Fig. 7, the compound expressions 'SadSur', 'SadFear', 'FearSur', 'SadDis', 'AngSur', and 'DisSur' represent 'Sadly Surprised', 'Sadly Fearful', 'Fearfully Surprised', 'Sadly Disgusted', 'Angrily Surprised', and 'Disgustedly Surprised', respectively. We can see that a compound expression has higher similarities with its relevant basic expressions than with other basic expressions.

Visualization of Confusion Matrices. Fig. 8 visualizes the confusion matrices obtained by different methods at the last incremental task on CFEE (C=5). We can observe that the Baseline method has a strong bias towards new classes and forgets the previously learned knowledge. SCN is also prone to overfit new classes, since the model may relabel the samples of old classes to new classes. FOSTER is prone to fit old classes, and thus its plasticity is not as good as its stability. Among all the competing methods, our method strikes a better balance between old and new classes.

Visualization of Expression Features. In Fig. 9, we visualize the extracted expression features of test data by t-SNE on RAF-DB (C=3) and CFEE (C=3). Compared with the features obtained by Baseline, the features obtained by our method give a better feature distribution (i.e., the intra-class compactness is enhanced while inter-class separability is enlarged). This shows our method can effectively learn discriminative features for identifying different expressions.

E. Comparison with State-of-the-Art Methods

Table V shows the performance comparisons between our method and state-of-the-art methods (including CIL methods [14]–[16], [26], [41] and a representative FER method [5]) with the different numbers of incremental classes on RAF-DB,

TABLE V
PERFORMANCE COMPARISONS (THE AVERAGE ACCURACY (%) AND THE STANDARD DEVIATION OVER THE INCREMENTAL TASKS) BETWEEN OUR PROPOSED METHOD AND SEVERAL STATE-OF-THE-ART METHODS WITH THE DIFFERENT NUMBERS OF INCREMENTAL CLASSES C=3 AND C=5 ON CFEE, RAF-DB, AND EMOTIONET. THE BEST RESULTS ARE MARKED IN **BOLD**.

Methods	Params	FLOPs	CFEE		RAF-DB		EmotioNet	
			C=3	C=5	C=3	C=5	C=3	C=5
iCaRL [15]	11.18M	1.82G	$67.39_{\pm 1.25}$	$68.27_{\pm 1.64}$	$63.33_{\pm 0.79}$	$63.96_{\pm 0.22}$	$59.48_{\pm 0.44}$	$61.40_{\pm 0.77}$
PODNet [14]	11.18M	1.82G	$63.82_{\pm 1.85}$	$66.31_{\pm 1.55}$	$58.36_{\pm 1.20}$	$61.02_{\pm 0.92}$	$56.11_{\pm 0.57}$	$59.73_{\pm 1.32}$
COIL [41]	11.18M	1.82G	$56.35_{\pm 1.26}$	$58.25_{\pm 0.47}$	$47.73_{\pm 2.65}$	$48.34_{\pm 1.13}$	$52.85_{\pm 2.21}$	$56.38_{\pm 1.62}$
AFC [26]	11.18M	1.82G	$65.54_{\pm 1.75}$	$66.81_{\pm 1.49}$	$68.59_{\pm 1.11}$	$66.96_{\pm 0.47}$	$59.79_{\pm 1.50}$	$61.75_{\pm 0.91}$
FOSTER [16]	22.35M	3.65G	$62.12_{\pm 1.60}$	$62.39_{\pm 1.17}$	$69.11_{\pm 0.58}$	$70.04_{\pm 0.27}$	$60.90_{\pm 2.06}$	$62.85_{\pm 0.40}$
MEMO [42]	44.75M	3.47G	$66.01_{\pm 2.28}$	$67.95_{\pm 1.97}$	$63.22_{\pm 1.47}$	$62.49_{\pm 0.72}$	$57.87_{\pm 1.85}$	$58.73_{\pm 0.93}$
SCN [5]	11.18M	1.82G	$46.62_{\pm 0.23}$	$53.73_{\pm 1.29}$	$43.34_{\pm 2.53}$	$40.34_{\pm 1.45}$	$50.21_{\pm 1.84}$	$55.40_{\pm 1.43}$
Baseline	11.18M	1.82G	$59.51_{\pm 1.59}$	$60.91_{\pm 1.29}$	$44.12_{\pm 1.15}$	$45.13_{\pm 0.69}$	$53.92_{\pm 0.92}$	$55.91_{\pm 3.03}$
RGKT (Ours)	11.28M	1.84G	**68.44**$_{\pm 2.56}$	**68.87**$_{\pm 1.99}$	**70.34**$_{\pm 0.96}$	**71.91**$_{\pm 0.43}$	**61.66**$_{\pm 1.13}$	**63.27** $_{\pm 0.67}$

CFEE, and EmotioNet. Moreover, we visualize the comparison results on RAF-DB in Fig. 10. Note that existing compound FER methods [10], [11] do not release their source codes or models. Thus, they are not taken for performance comparisons.

The Baseline method can adapt to new classes but it cannot remember the learned knowledge well. FOSTER gives good performance on old expressions but obtains low accuracy on new expressions. iCaRL, PODNet, and AFC explore different distillation strategies, where iCaRL leverages a distillation loss via old exemplars while PODNet and AFC utilize the distillation loss to prevent the model from forgetting important information of old classes. Both COIL and our method leverage knowledge transfer. However, their differences are significant in terms of motivation and methodology. COIL develops a semantic mapping to transfer old classifiers to new classes with the optimal transport and transfer new classifiers to old classes symmetrically. In contrast, we initialize the new classifiers based on the expression relevance between new compound expressions and basic expressions (involving a relatively small amount of computation burden). Meanwhile, we alleviate the imbalance between old and new classes by exploiting expression-specific knowledge. Note that the semantic mapping in COIL does not fit FER CIL very well. For the FER method, SCN achieves good accuracy on the classification of basic expressions but fails to identify old expressions in the incremental tasks. Although the distillation loss and exemplars from old classes are used to train SCN, it still suffers from catastrophic forgetting since the bias towards new classes makes the model easily relabel new classes.

Existing methods ignore the importance of the relationship across expressions in the class-incremental FER. In contrast, our RGKT method explores this relationship from two perspectives (i.e., compound expression-oriented knowledge and basic expression-oriented knowledge). In such a way, RGKT initializes the new class classifiers based on expression relevance between compound and basic expressions. Meanwhile, it learns expression-generic knowledge and transfers this knowledge into old classes. This effectively avoids the limited information of old exemplars. Among all the competing methods, RGKT achieves the best performance in terms of average accuracy on three databases and is comparable in both the number of parameters and FLOPs. Specifically, RGKT obtains the highest accuracy of 68.44% (68.87%) on the in-the-lab CFEE database, 70.34% (71.91%), and 61.66% (63.27%) on the in-the-wild RAF-DB and EmotioNet databases, respectively, when the number of incremental classes is C=3 (5). In general, RGKT can effectively balance the trade-off between old and new expressions. Although our method performs the best among the competing methods on three databases, the number of expression-generic features is fixed, which may limit the plasticity of new classes.

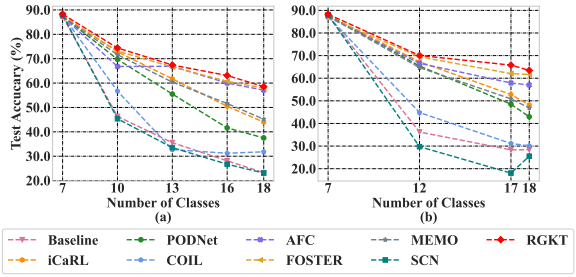

Fig. 10. **Test accuracy** for (a) C=3 and (b) C=5 w.r.t. the number of classes obtained by different methods on RAF-DB.

V. CONCLUSION AND FUTURE WORK

In this paper, we study a novel and practical setting for class-incremental FER, where we take well-studied and easily-accessible basic expressions as initial classes and identify new compound expressions incrementally. By effectively exploiting the intrinsic relationship across expressions, we design an RGKT method (consisting of an MFL module, a BET module, and a CET module) specifically suited for this setting. The MFL module captures subtle distinctions in expressions. This not only improves the discriminative ability of our model to classify various expressions, but also facilitates accurate knowledge transfer for the incremental task. The BET and CET modules largely alleviate the stability-plasticity dilemma by transferring expression-related knowledge, thereby adapting to new classes and relieving the forgetting of old classes. Extensive experiments demonstrate the superior performance of our method against several state-of-the-art methods.

Currently, we leverage the rehearsal strategy to store a tiny set of exemplars of old classes. In future work, we will further study non-exemplar class-incremental FER by exploring the relationship between expressions and the distribution information of old classes during incremental learning.

REFERENCES

[1] J. Cai, Z. Meng, A. S. Khan, Z. Li, J. O'Reilly, and Y. Tong, "Island loss for learning discriminative features in facial expression recognition," in *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, 2018, pp. 302–309.

[2] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2852–2861.

[3] D. Ruan, Y. Yan, S. Lai, Z. Chai, C. Shen, and H. Wang, "Feature decomposition and reconstruction learning for effective facial expression recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7660–7669.

[4] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao, "Region attention networks for pose and occlusion robust facial expression recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 4057–4069, 2020.

[5] K. Wang, X. Peng, J. Yang, S. Lu, and Y. Qiao, "Suppressing uncertainties for large-scale facial expression recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6897–6906.

[6] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion." *Journal of Personality and Social Psychology*, vol. 17, no. 2, pp. 124–129, 1971.

[7] S. Du, Y. Tao, and A. M. Martinez, "Compound facial expressions of emotion," *Proceedings of the National Academy of Sciences*, vol. 111, no. 15, pp. E1454–E1462, 2014.

[8] X. Zou, Y. Yan, J.-H. Xue, S. Chen, and H. Wang, "When facial expression recognition meets few-shot learning: A joint and alternate learning framework," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 5367–5375.

[9] C. Fabian Benitez-Quiroz, R. Srinivasan, and A. M. Martinez, "EmotioNet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5562–5570.

[10] H. Li and Q. Li, "End-to-end training for compound expression recognition," *Sensors*, vol. 20, no. 17, p. 4727, 2020.

[11] K. Slimani, K. Lekdioui, R. Messoussi, and R. Touahni, "Compound facial expression recognition based on highway CNN," in *Proceedings of the New Challenges in Data Sciences: Acts of the Second Conference of the Moroccan Classification Society*, 2019, pp. 1–7.

[12] R. M. French and N. Chater, "Using noise to compute error surfaces in connectionist networks: A novel means of reducing catastrophic forgetting," *Neural Computation*, vol. 14, no. 7, pp. 1755–1769, 2002.

[13] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 233–248.

[14] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "PODNet: Pooled outputs distillation for small-tasks incremental learning," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 86–102.

[15] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[16] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "FOSTER: Feature boosting and compression for class-incremental learning," in *Proceedings of the European Conference on Computer Vision*, 2022, pp. 398–414.

[17] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 374–382.

[18] S. Yan, J. Xie, and X. He, "DER: Dynamically expandable representation for class incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3014–3023.

[19] Z. Ji, Z. Hou, X. Liu, Y. Pang, and X. Li, "Memorizing complementation network for few-shot class-incremental learning," *IEEE Transactions on Image Processing*, vol. 32, pp. 937–948, 2023.

[20] S. Grossberg, "Adaptive resonance theory: How a brain learns to consciously attend, learn, and recognize a changing world," *Neural Networks*, vol. 37, pp. 1–47, 2013.

[21] R. Mo, Y. Yan, J.-H. Xue, S. Chen, and H. Wang, "D^3Net: Dual-branch disturbance disentangling network for facial expression recognition," in *Proceedings of the ACM International Conference on Multimedia*, 2021, pp. 779–787.

[22] D. Ruan, Y. Yan, S. Chen, J.-H. Xue, and H. Wang, "Deep disturbance-disentangled learning for facial expression recognition," in *Proceedings of the ACM International Conference on Multimedia*, 2020, pp. 2833–2841.

[23] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 532–547.

[24] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[25] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "Overcoming catastrophic forgetting by incremental moment matching," in *Advances in Neural Information Processing Systems*, 2017, pp. 1–11.

[26] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 071–16 080.

[27] K. Li, J. Wan, and S. Yu, "CKDF: Cascaded knowledge distillation framework for robust incremental learning," *IEEE Transactions on Image Processing*, vol. 31, pp. 3825–3837, 2022.

[28] C.-Y. Hung, C.-H. Tu, C.-E. Wu, C.-H. Chen, Y.-M. Chan, and C.-S. Chen, "Compacting, picking and growing for unforgetting continual learning," *Advances in Neural Information Processing Systems*, 2019.

[29] J. Zhu, B. Luo, S. Zhao, S. Ying, X. Zhao, and Y. Gao, "IExpressNet: Facial expression recognition with incremental classes," in *Proceedings of the ACM International Conference on Multimedia*, 2020, pp. 2899–2908.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

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

[32] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 831–839.

[33] M. Yang, D. He, M. Fan, B. Shi, X. Xue, F. Li, E. Ding, and J. Huang, "DOLG: Single-stage image retrieval with deep orthogonal fusion of local and global features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 772–11 781.

[34] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9719–9728.

[35] B. H. Ali Mollahosseini, M. H. Mahoor, and M. H. Mahoor, "AffectNet: A database for facial expression, valence, and arousal computing in the wild," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 18–31, 2017.

[36] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, and D.-C. Zhan, "PyCIL: A Python toolbox for class-incremental learning," *arXiv preprint arXiv:2112.12533*, 2021.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[38] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

[39] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[40] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE." *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[41] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Co-transport for class-incremental learning," in *Proceedings of the ACM International Conference on Multimedia*, 2021, pp. 1645–1654.

[42] D.-W. Zhou, Q.-W. Wang, H.-J. Ye, and D.-C. Zhan, "A model or 603 exemplars: Towards memory-efficient class-incremental learning," in *Proceedings of International Conference on Learning Representations*, 2023, pp. 1–14.