We provide more details and results about our work in the appendices. Here are the contents:

- Appendix A: Details of Granular Ball Represent.
- Appendix B: Additional Experiment Results.

A Details of Granular Ball Represent

In this section, we present in more detail the whole process of Granular Ball Represent (GBR) for expression images.

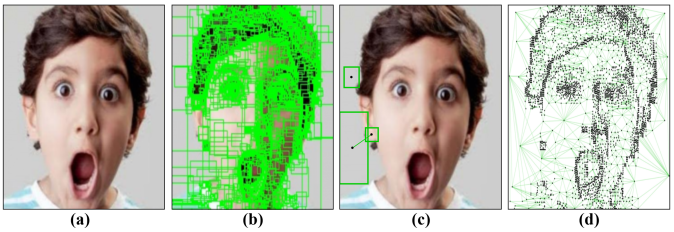


Figure 7: Process decomposition of GBR . (a) original image, (b) the intermediate process figure of GBR, (c) the simplification of (b), (d) the final result of GBR.

A.1 Fundamental idea of GBR

Given an image of expression x^i, by using Algorithm 1 [Xia *et al.*, 2023], we can obtain the GB_list, which is a list containing several matrices. Previously we have been talking about the field of granular balls, but the specific implementation is in the form of granular matrix. This is due to in two-dimensional surface, the size of images are determined by x, y, while the granular ball can only rely on the radius to determine the size of GBR. In [Xia *et al.*, 2023], while innovatively applying GBR to images, Xia $et\,al.$ proposed the granular moment as a unit of representation which is more suitable for image data. The GB_list represents the visualization on x^i as shown in Fig. 7b. Each node in the graph x^g corresponding to x^i is centered by the corresponding matrix element within GB_list. For each edge of x^g, it consists of the line joining the centroids of any two matrices with overlapping regions, as shown in Fig. 7c.

A.2 Algorithm of GBR

Algorithm 1 algorithm of GBR

Input: an image of expression: x^i
Parameter: P_{thr}, thr_1, Var_{thr}
Output: GB_list

Computing the gradient of each pixel p_j in x^i: x^i_{grad}
Set label $l(p_j) = 0$ for each pixel p_j in x^i_{grad}
Set $x^i_{grad} = x^i_{grad}.sort$
repeat
  /* Select the center point with minimal gradient.*/
  if $\exists l(p_j) == 0, p_j \in x^i_{grad}$ **then**
    $[grad_m in, c_k] = x_{grad}.min$
  end if
  /* Rectangular region searching, $(2 * r_x + 1, 2 * r_y + 1)$ represents (width, length) and $f(p_j)$ represents the gray value of p_j. */
  Set $r_x = 0, r_y = 0$
  repeat
    $(r_x += 1\ or\ r_y += 1) \rightarrow Region\ R_k$
    $Purity_k = 1 - \frac{\sum(f(p_j)-f(c_k))>thr_1}{(2*r_x+1)(2*r_y+1)}, j \in R_k$
    Compute the variance of R_k: Var_k
    if $Purity_k < P_{thr} \| Var_k > Var_{thr}$ **then**
      if $r_x(r_y)$ plus one in this iteration. **then**
        $r_x -= 1 (r_y -= 1)$ and $r_x(r_y)$ stop increasing in later iterations.
      end if
    end if
    Set $P_{thr} * = 1.005$
  until $r_x \& r_y\ stop\ increasing.$
  /* Update the graph.*/
  $GB_k = (c_k, r_x, r_y, Purity_k, Var_k, Mean_k)$
  Add GB_k to GB_list.
  $l(p_j) = 1$ for p_j in R_k
until $l(p_j) == 1$ for each pixel p_j in x_{grad}
return GB_list

A.3 Features of Node and Edge

We tried to design more features related to spatial information from the perspective of edge points, expecting to provide more spatial information to the model.

Features of nodes. This part of the features consists of the position coordinates (x, y) of the Node, the variance Var_k, mean $Mean_k$, skewness $Skew_k$, maximum Max_k and minimum Min_k of the $k-th$ matrix region, and the entropy Ent_k which can reflect the complexity of the information in the matrix region, which can be formulated as:

$$Fea_n = (x, y, Var_k, Mean_k, Skew_k.Max_k, Min_k, Ent_k), \tag{11}$$

where Fea_n denotes the features of node.

Features of edges. This part of the features consists of the length l of the edge, the positional coordinates of the two end nodes $(x1, y1, x2, y2)$, and the area s of the overlap region, which can be formulated as:

$$Fea_e = (l, x1, y1, x2, y2, s), \tag{12}$$

where Fea_e denotes the features of edge.

A.4 Details of Graph Down Sample

The graph generated by GBR contains a large number of nodes, some of which are located in the hair, clothing, and other non-expression related areas, which will import noise into the model. This will reduce the accuracy of the model if it is not processed. To address this problem, we perform graph downsampling in GBR to retain spatial information while minimizing the number of noise points. As shown in Fig. 8, the number of noise points is drastically reduced when 10% of nodes is retained. At the same time, the AGM and $\mathcal{L}_{BA}$ modules constructed by our method enable the visual representation to guide the attention region of the spatial representation, which further weakens the negative impact of irrelevant regions.

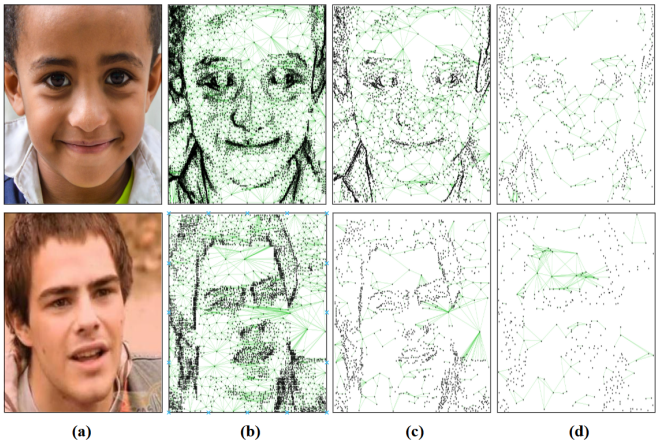


Figure 8: Different downsampling ratios of GDS. (a) original image, (b) Retain 100% of nodes, (c) Retain 50% of nodes, (d) Retain 10% of nodes.

The number of nodes contained in x^g is closely related to the resolution of x^i as well as the tonal complexity. And high resolution images are very common these days. The downsample operation of [Xia *et al.*, 2023] is only dedicated to reduce the nodes with edge connections. In order to effectively reduce the redundant nodes in x^g, we design a downsample operation for edge-connectionless nodes, the basic idea is that for $j-th$ node n_j in x^g which has no edge-connections, the node merging is performed by taking the nearest node n_k, where the distance between the two points needs to be less than a threshold value of d. The two downsample operations are used together in the process of x^g generation. The number of nodes per graph is reduced to within the number N. The value of d is determined by the resolution $(W \times H)$ as well as N, which can be formulated as:

$$d = \frac{W}{(\sqrt{N}-1)}, \; when \, W = H \tag{13}$$

where d is the threshold.

For easier explanation, Eq. 13 simplifies the issue by assuming that W and H are equal. The d derived from Eq. 13 is a limiting case, and it is impossible to have a situation where the distance between all nodes is greater than d and the number of summarized points is still greater than N. d ensures that the total number of nodes after final downsample is within the range N . Our method selects the merge of the nearest nodes, which greatly maintains the quality of the graph data while reducing the number of nodes. The following is the same theoretical reasoning when W and H are not equal, which can be formulated as:

$$\begin{cases} n_w = \frac{n_h \times W}{H} \\ f(n_h) : (\frac{n_h \times W}{H}) \times n_h <= N \\ n_h = argmax(f(n_h) \end{cases} \tag{14}$$

$$d = max(\frac{W}{n_w}, \frac{H}{n_h}) \tag{15}$$

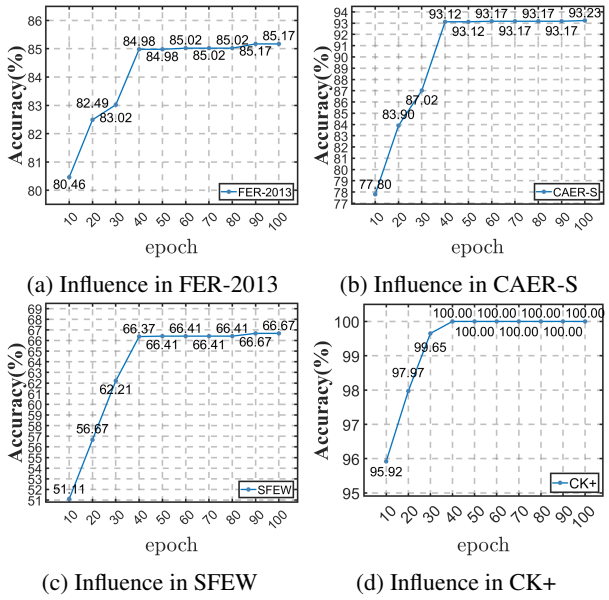
B Additional Experiment Results

B.1 Ablation of Others Setting

In this subsection, we provide ablation studies for the remaining model parameter settings.

downsampling ratios of GDS	*Accuracy (%)*	
	CAER-S	CK+
100%	92.06%	98.86%
50%	92.76%	99.43%
10%	**93.12%**	**100.00%**
5%	92.67%	99.39%

Table 5: The ablation study of different downsampling ratios in the process of GDS. Evaluating on the validation set of the databases.



(a) Influence in FER-2013 (b) Influence in CAER-S

(c) Influence in SFEW (d) Influence in CK+

Figure 9: Ablation studies for the different epoch of training on validation set.

Influence of different downsampling ratios. We evaluate the recognition performance of the proposed method with the different downsampling ratios in the process of GDS, as shown in Table 5. The graphs generated by GBR contain spatial information that images do not have. However, the introduced spatial information contains not only valid spatial information but also noise spatial information. We set the appropriate downsampling rate to eliminate the noise as much as possible while introducing the spatial information. Specially, we can observe that the model achieves the best results with the downsampling rate set to 10%.

Influence of the epoch of training. We evaluate the recognition performance of the proposed method with the different epochs in the process of training, as shown in Fig. 9. Specially, we can Observe that the proposed method achieves a nice recognition accuracy before 40 epochs of training. After that, with the epoch increasing, the recognition accuracy can hardly be greatly improved but the time required for its training increases dramatically.

Influence of the batch size of training. We evaluate the recognition performance of the proposed method with the different batch sizes in the process of training, as shown in

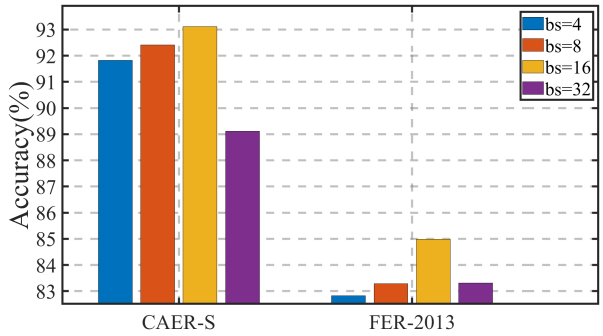
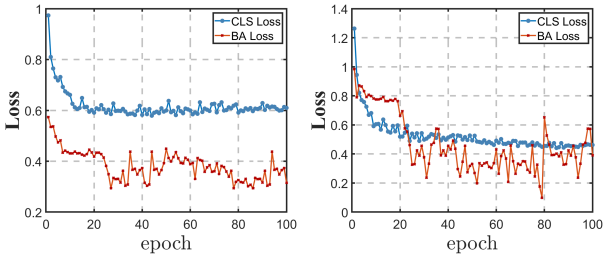


Figure 10: Ablation studies for the different batch size of training on the CAER-S and FER-2013 databases. Note that bs means the batch size of training.



(a) Influence in FER-2013 (b) Influence in RAF-DB

Figure 11: Experiments of convergence analysis on different databses.

Fig. 9. We can note that the proposed method achieves the best accuracy when the batch size is set to 16. Too small a batch size causes the model to focus too much on the differences between batches of samples during training to update the training parameters frequently, which makes the gradient more stochastic. Increasing the batch size helps with convergence stability, but the generalization performance of the model decreases as the batch size increases.

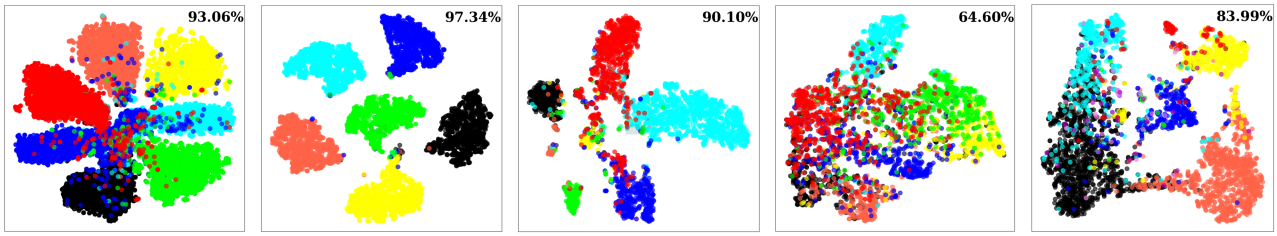
B.2 Convergence Analysis

In this subsection, convergence analysis experiments are performed on the databases used for our experiments. As shown in Fig. 11, the increasing number of epochs leads to a gradual convergence of all loss functions, ultimately reaching a stable state. This clear observation serves as compelling evidence for the stability and effectiveness of our proposed model.

B.3 More Visualization

2D feature visualization. We use t-SNE to visualize F_u of CS-SBF on different test databases on the 2D space, respectively, as shown in Fig. 12. With t-SNE visualization, we can conclude that the proposed method has good classification results not only within the network. Visualized on the 2D space, we can intuitively see that F_u can effectively reduce the intra-class differences and enhance the inter-class separability of different expressions.

More Attention visualization. In this subsection, we provide more attention visualization of different expression images, as shown in Fig. 13.



(a) CAER-S (b) Oulu-CASIA (c) RAF-DB (d) AffectNet-7 (e) FER-2013

Figure 12: Visualization of fusion features F_u using t-SNE.

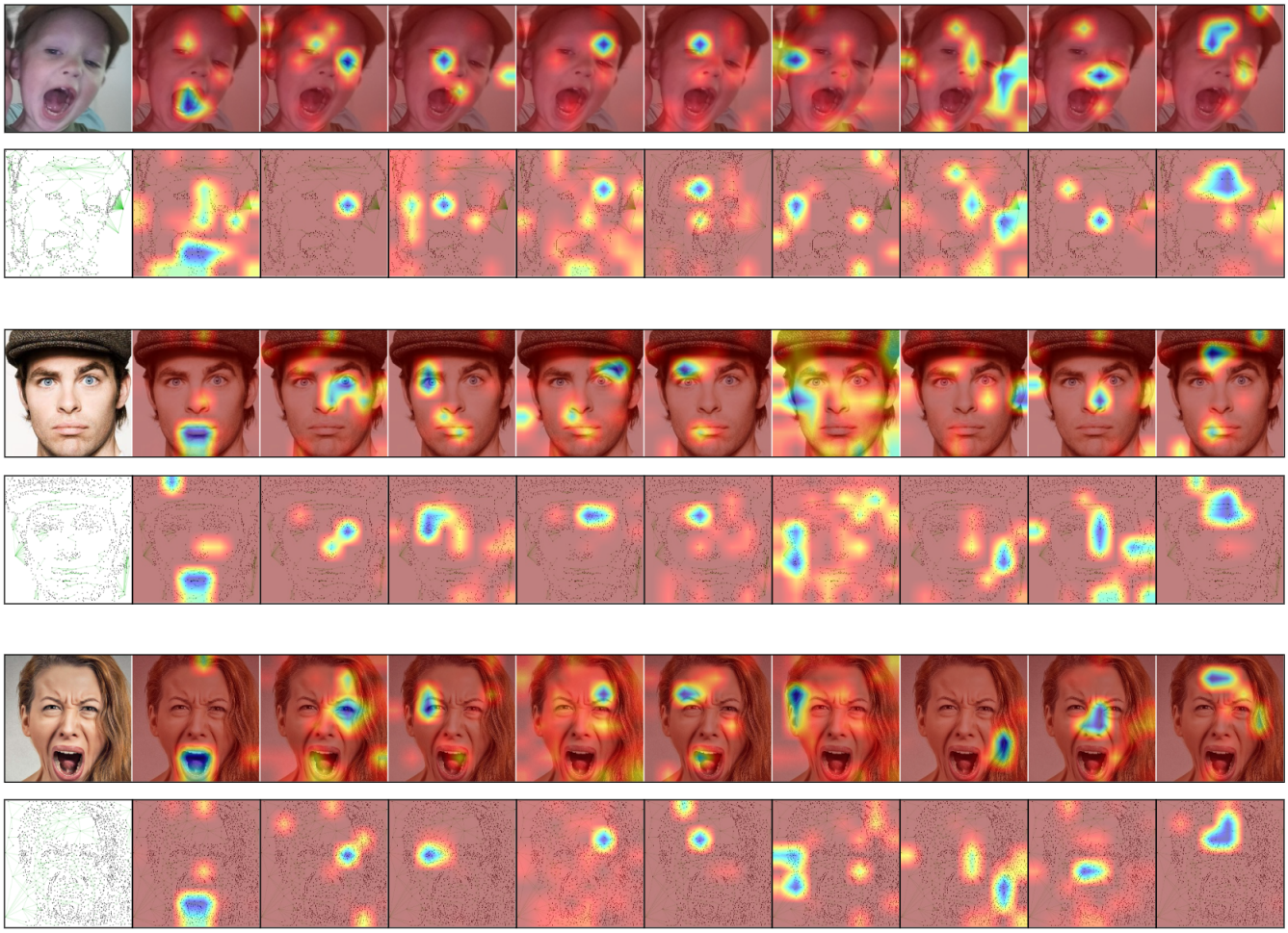


Figure 13: Visualization of component representation features C^v and C^s by CAM on more samples.