# Learning to Cluster Faces with Mixed Face Quality

Zhiwei Pan[1,2(✉)], Zhihua Guo[3], Huiting Yang[4], Congquan Yan[1], and Pengju Yang[1]

[1] Hikvision Research Institute, Hangzhou, China
{panzhiwei5,yancongquan,yangpengju}@hikvision.com
[2] Zhejiang University, Hangzhou, China
[3] Xiamen University, Xiamen, China
31520191153352@stu.xmu.edu.cn
[4] Nanjing University of Aeronautics and Astronautics, Nanjing, China
yanghuiting@nuaa.edu.cn

**Abstract.** Face clustering is the task of grouping faces by their underlying identity, and is still a challenging task in practical use due to the common low-quality face images caused by pose, blur, occlusion, illumination etc. To address the issue, this paper proposes a face clustering algorithm, referred as FC-Q, that takes the quality score as extra input. Based on the main observation that two nodes similar in feature subspace but with different identity may have larger score difference, the algorithm first integrates this prior with the modified self-attention mechanism of Transformer to infer reliable linkage likelihood between similar node pairs. Then the algorithm combines the face quality information with the label propagation module to further suppress the abnormal pairings. The effectiveness of the algorithm is evaluated on two public face datasets in good and bad quality. Experimental results validate that our algorithm outperforms the state-of-the-arts under the general circumstance of clustering faces with mixed face quality.

**Keywords:** Face clustering · Face quality · Linkage estimation · Label propagation · Self-attention mechanism

## 1 Introduction

Face clustering is a fundamental task in face analysis and has been extensively studied in resent years [8,15,20,22,23]. Existing face clustering methods roughly fall into two categories, i.e., unsupervised methods and supervised methods. Unsupervised approaches, such as K-Means [10] and DBSCAN [6], rely on specific assumptions and lack the capability of resonating with high-dimensional structured data information. Supervised face clustering methods mainly aim to

This work was supported by the Postdoctoral Merit Funding Project of Zhejiang Province under Grant ZJ2020050.


S. Yu et al. (Eds.): PRCV 2022, LNCS 13535, pp. 576–587, 2022.
https://doi.org/10.1007/978-3-031-18910-4_46

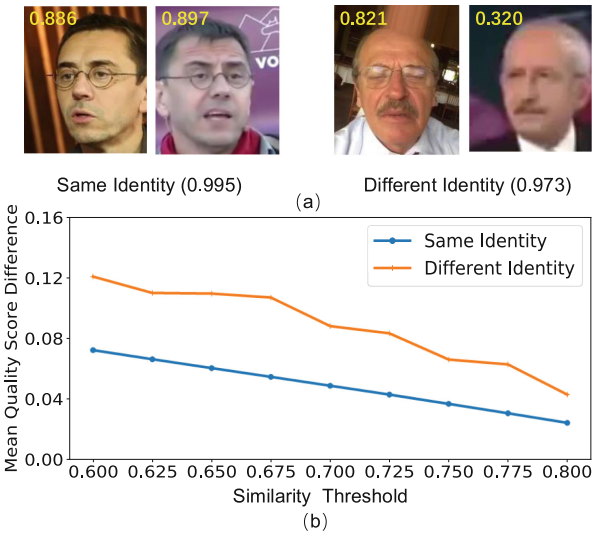


**Fig. 1.** (a) Demonstration of two node pairs with quality score shown on the images and similarity given in the parentheses. (b) Mean absolute value of quality score difference between every two pair nodes on IJB-C dataset [11] under the same and different identities with respect to similarity threshold. The node pairs with similarity higher than threshold are involved in this statistic. The face score is obtained by EQFace [9], the identity feature is extracted by pre-trained IResNet50 [4,5].

learn more distinguishing embedding subspace [3,14] or the complex cluster patterns [23]. These existing methods mainly based on the node distance in feature space, while ignore the negative effect caused by the face images in low quality. Face node with low quality is common and goes against the clustering due to its ambiguous identity. Figure 1(a) shows the example that the pair nodes with one in low quality can also have high similarity even they are under different identity. These low face quality nodes will obviously degrade the face clustering precision if not handled appropriately. It is essential for face clustering algorithms to have the ability to deal with this general application circumstance.

Fortunately, face quality score provides helpful auxiliary information for clustering. Figure 1(b) shows the mean absolute value of quality score difference between every two pair nodes under same and different identities with respect to similarity threshold. The node pairs in IJB-C dataset [11] with similarity higher than threshold are involved in this statistic. There always exists a gap between the score differences, which implies that two nodes similar in feature subspace but with different identity may have larger score difference. Based on this main observation, this paper proposes a face clustering algorithm, which is referred as FC-Q, intuitively takes the face quality as extra input to exploit unlabeled face data.

Face quality can be conveniently assessed beforehand by recent deep learning based methods. SER-FIQ [17] obtains the face quality by measuring the

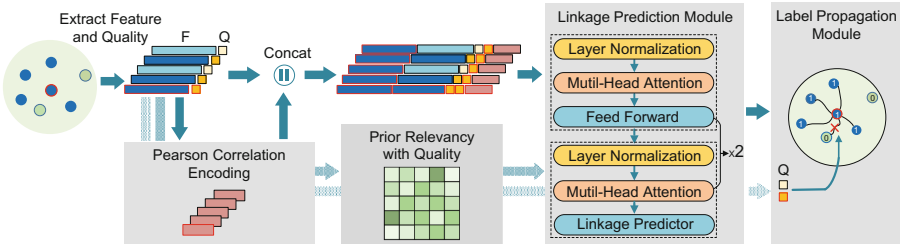


**Fig. 2.** Flowchart of our FC-Q algorithm. The proposed algorithm follows the link-based clustering paradigm, Given one pivot face node and its $K$ nearest neighbors in feature subspace, the proposed algorithm first obtains the input vectors by concatenating the identity features and face scores of pivot-neighbor pairs as well as the specific Pearson correlation encoding. Its linkage prediction module is in the form of modified Transformer encoder, and infers linkage likelihood with the help of pre-calculated prior relevancy matrix. Its label propagation module transitively merges face nodes according to the linkage likelihood refined with local face quality information. Instances with the same pseudo label constitute a cluster.

embedding variations generated from random sub-networks of the face recognition model. A deep tiny network [13] is also proposed to learn a face quality prediction function that is recognition-oriented. Meanwhile, face quality can be explicitly given along with the identity feature by face recognition network. MagFace [12] introduces an adaptive mechanism to learn a universal feature embedding with magnitude measuring the face quality. EQFace [9] outputs face quality and identity feature at the same time by adding a quality network branch to the baseline network of face recognition. Such methods make our work more efficient in gathering input data.

The proposed FC-Q algorithm incorporates the face quality into its two main modules, i.e., the linkage prediction module and the label propagation module. In the linkage prediction module, the algorithm adopts the framework of Transformer encoder [18]. It is specially modified to fit the general clustering circumstance. One relevancy prior is designed according to the quality score relationship among nodes in neighborhood. The prior helps the self-attention mechanism of Transformer encoder better infer the linkage likelihood between node pairs. In the label propagation module, the algorithm utilizes the face quality to recalibrate the abnormal linkage likelihood based on the local quality information. The linkage with one node having inconsistent quality score with its neighbors will be suspected of being unreliable, and its likelihood will be suppressed if the two pair nodes have a large gap with respect to their local quality information. Finally, our proposed algorithm transitively merges face nodes according to the refined linkage likelihood, and obtains the clusters.

To summarize, the main contributions of this work are as follows:

– A face clustering algorithm named FC-Q is proposed with the face quality as extra input. Compared with the state-of-the-arts, this algorithm deals with

the more general circumstance that the face nodes are not all guaranteed in good quality.
- The proposed FC-Q algorithm modifies the Transformer encoder and designs the relevancy prior with face quality to infer more reliable linkage likelihood between similar pairs in feature subspace.
- The proposed FC-Q algorithm utilizes the local quality information of each node to further suppress the pairing with abnormal high linkage likelihood.
- The proposed FC-Q algorithm specifically conducts face clustering experiments on IJB-C dataset with low face quality and achieves 91.7% *pairwise* F-score on partial IJB-C, which provides a strong baseline for low-quality face clustering.

## 2 Methodology

In this section, we introduce the details of the proposed FC-Q algorithm, which includes the specific linkage prediction module and label propagation module. Figure 2 shows the flowchart.

### 2.1 Linkage Prediction with Prior Attention

Following the link-based clustering paradigm, the proposed algorithm selects every face node as a pivot, and estimates the linkage likelihood between the pivot and its $K$ nearest neighbors in feature subspace.

Given $i$th pivot face node $\mathbf{f}_0^i \in \mathbb{R}^{d_f}$ and its $K$ nearest neighbors $\mathbf{f}_j^i, j \in \{1, ..., K\}$, the input of the linkage prediction module consists of $K+1$ vectors with each has the form

$$\mathbf{g}_j = \text{cat}\left(\mathbf{f}_j^i, \mathbf{f}_0^i, q_j^i, q_0^i, \mathbf{p}_j\right) \in \mathbb{R}^{d_g}, \quad j \in \{0, ..., K\}, \tag{1}$$

where $q_0^i \in \mathbb{R}$ denotes the face quality score of pivot node, and $q_j^i, j \in \{1, ..., K\}$ denotes the face quality score of its neighbor. Operator cat() denotes concatenation operation along the feature dimension, thus $d_g = 2d_f + 2 + K + 1$. Vector $\mathbf{p}_j \in \mathbb{R}^{K+1}$ represents Pearson correlation encoding of each involved face node. The element in $\mathbf{p}_j \in \mathbb{R}^{K+1}$, i.e., the Pearson correlation coefficient, is calculated as

$$\mathbf{p}_j^k = \frac{\text{cov}\left(\mathbf{f}_j^i, \mathbf{f}_k^i\right)}{\sigma_{\mathbf{f}_j^i}\sigma_{\mathbf{f}_k^i}}, \quad k \in \{0, ..., K\}, \tag{2}$$

where cov and $\sigma$ denote the covariance and the standard deviation of $\mathbf{f}_j^i$ and $\mathbf{f}_k^i$ respectively. Concatenating pivot feature $\mathbf{f}_0^i$ in (1) aims to inform the linkage prediction module to learn the relationship between pivot-neighbor pair. Concatenating quality scores $q_j^i, q_0^i$ in (1) aims to let the linkage prediction module infer the link likelihood with more references. Concatenating Pearson correlation encoding $\mathbf{p}_j$ in (1) is under the consideration that the Pearson correlation coefficient can offer the linkage prediction module robust linear relations between pair nodes, which is highly sensitive to outliers [2].

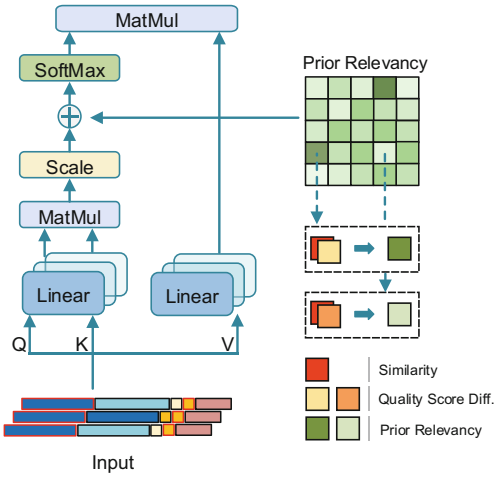


**Fig. 3.** Architecture of our modified self-attention mechanism. The input vectors are first projected into three super vectors named *key*, *query* and *value*. The *key* and *query* are forced to share the same projection. A prior relevancy matrix is added into the learned relevancy, and offers the negative relevancy information if the pair nodes have large quality score difference. The brighter the color, the larger the value.

With the input vectors generated, the linkage prediction module adopts the framework of modified Transformer encoder to estimate the linkage likelihood of pivot-neighbor pair. Compared with the GCN based framework [20], the Transformer encoder framework has the ability to learn relation weights based on its effective self-attention mechanism. As shown in Fig. 2, the linkage prediction module is composed of a stack of 3 identical layers. Each layer has three sub-layers. The first is a pre-layer normalization [21], the second is a modified multi-head self-attention mechanism, and the third is a simple fully connected feed-forward network. Note that the skip connections of the last two sub-layers are specifically removed to let the whole module focus more on inferring the difference between pivot and neighbor nodes. The last feed-forward sub-layer performs the binary node classification followed by softmax activation, which outputs the probability of whether the corresponding input belongs to the same class as the pivot.

Figure 3 further shows the architecture of our three modified self-attention mechanism sub-layers. Taking the first one as an example, every normalized vector $\mathbf{g}'_j$ in $i$th input $\mathbf{G}'_i \in \mathbb{R}^{(K+1)\times d_g}$ is first linearly projected into three super vectors named *key*, *query* and *value*, which can be expressed in matrix form as

$$\begin{aligned} \mathbf{K} &= \mathbf{G}'_i\mathbf{W}^S, \mathbf{K} \in \mathbb{R}^{(K+1)\times d_s}, \\ \mathbf{Q} &= \mathbf{G}'_i\mathbf{W}^S, \mathbf{Q} \in \mathbb{R}^{(K+1)\times d_s}, \\ \mathbf{V} &= \mathbf{G}'_i\mathbf{W}^V, \mathbf{V} \in \mathbb{R}^{(K+1)\times d_v}, \end{aligned} \tag{3}$$

where matrices $\mathbf{W}^S \in \mathbb{R}^{d_g \times d_s}$ and $\mathbf{W}^V \in \mathbb{R}^{d_g \times d_v}$ denote the learnable project matrices. The *key* and *query* are forced to share the same projection, thus the relevancy between the corresponding two samples will be symmetric. This setting is helpful for face clustering, which is shown in the following experiment section.

The relevancy between the samples are constructed as

$$\frac{1}{\sqrt{d_s}}\mathbf{Q}\mathbf{K}^T + \mathbf{A}, \tag{4}$$

where matrix $\mathbf{A} \in \mathbb{R}^{(K+1)\times(K+1)}$ represents the prior relevancy, and its element in the $m$th row and $n$th col is of the form

$$\mathbf{A}_{m,n} = \mathrm{sim}\left(\mathbf{f}_m^i, \mathbf{f}_n^i\right) \cdot \left(1 - \mathrm{abs}(q_m^i - q_n^i)\right), \tag{5}$$

where operator sim() calculates similarity between the identity features $\mathbf{f}_m^i$ and $\mathbf{f}_n^i$, and operator abs() denotes the absolute operation. Prior relevancy matrix $\mathbf{A}$ is also symmetric, and can offer the negative relevancy information between two samples if they have large quality score difference.

The output $\mathbf{Z} \in \mathbb{R}^{(K+1)\times d_v}$ in self-attention mechanism is the aggregation of *value* matrix $\mathbf{V}$ by attention weights, i.e.,

$$\mathbf{Z} = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T + \mathbf{A}}{\sqrt{d_s}}\right)\mathbf{V}, \tag{6}$$

where the attention weights are obtained by applying softmax normalization to (4).

The modified self-attention is further incorporated into the multi-head mechanism, where the self-attention is performed $H$ times in parallel. The outputs of each self-attention are concatenated and once again projected, resulting in the final output of multi-head self-attention sub-layer

$$\mathbf{Z}_M = \mathrm{cat}\left(\mathbf{Z}_1, ..., \mathbf{Z}_H\right)\mathbf{W}_M, \tag{7}$$

where matrix $\mathbf{W}_M \in \mathbb{R}^{Hd_v \times d_g}$ denotes learnable project matrix.

During the training stage, the whole linkage prediction module is trained by cross-entropy loss

$$\mathcal{L} = -\sum_{k=0}^{K} \log(\hat{y}_{k_1}^i), \tag{8}$$

where $\hat{y}_{k_1}^i$ denotes the output probability, i.e., linkage likelihood, of whether the $i$th pivot and its corresponding $k$th neighbor belong to the same class.

### 2.2 Label Propagation with Anomaly Suppression

As all the pivot nodes are involved in linkage prediction module, a set of pivot-neighbor pairs $\mathcal{E} = \{\mathbf{e}_j\}_{j=1,...,NK}$ will be obtained along with the corresponding linkage likelihood set $\mathcal{P} = \{p_j\}_{j=1,...,NK}$. The goal of the label propagation module is to assign pseudo label $y_i$ to every face image $\mathbf{x}_i$ with the help of quality score set.

**Algorithm 1.** Label propagation with anomaly suppression

**Input**: Pivot-neighbor pair set $\mathcal{E}$, linkage likelihood set $\mathcal{P}$, quality score set $\mathcal{Q}$.
**Parameter**: Initial likelihood threshold $\tau_p$, quality threshold $\tau_q$, maximum size $M$, maximum iteration number $T$.
**Output**: Pseudo labels.

Let $i = 0$.
**while** $i < T$ and $\mathcal{E} \neq \emptyset$ **do**
  **for** every pair $\mathbf{e}_j$ in $\mathcal{E}$ **do**
    Find connected neighbors of nodes $m$ and $n$ in $\mathbf{e}_j$.
    Calculate mean absolute value of quality score difference $\breve{q}_m$ and $\breve{q}_n$ using (9).
    **if** $\breve{q}_m < \tau_q$ or $\breve{q}_n < \tau_q$ **then**
      Suppress linkage likelihood using (10).
    **end if**
  **end for**
  Remove pairs from $\mathcal{E}$ with its likelihood below $\tau_p$.
  Find connected components.
  Annotate components with the node number below $M$ and remove its pairs from $\mathcal{E}$.
  Let $\tau_p = \tau_p + (1 - \tau_p) * 0.15$, and $i = i + 1$.
**end while**
Annotate orphan nodes.
**return** pseudo labels for all nodes

The label propagation module starts with the initial linkage threshold $\tau_p$, and performs in an iterative manner. In each iteration, the module first finds all the connected neighbors of every unlabeled node according to the current pair set $\mathcal{E}$. Then Given one pivot-neighbor pair $\mathbf{e}_j$ connecting nodes $m$ and $n$, the module calculates the mean absolute value of quality score difference between each node and its connected neighbors,

$$\breve{q}_* = \frac{1}{|\mathcal{N}_*|} \sum_{k \in \mathcal{N}_*} \text{abs}\left(q_* - q_k\right), \quad * = m, n, \tag{9}$$

where $\mathcal{N}_*$ denotes the set of neighbor index, and $|\mathcal{N}_*|$ denotes the number. If the value $\breve{q}$ of either node is larger than a predefined threshold $\tau_q$, the pivot-neighbor pair will be considered unreliable and its linkage likelihood will be suppressed as

$$p_j = p_j \cdot \left(1 - \text{abs}\left(\bar{q}_m - \bar{q}_n\right)\right), \tag{10}$$

where $\bar{q}_m$ and $\bar{q}_n$ denote the mean value of quality scores of the very node and its connected neighbors. The gap in local face quality information between the two nodes determines the degree to which the corresponding linkage likelihood is suppressed.

With the linkage likelihood all updated, the label propagation module first removes the pivot-neighbor pairs from $\mathcal{E}$ whose linkage likelihood are below the threshold $\tau_p$. Then the module finds connected components based on the remaining pivot-neighbor pairs. If the node number of one component is below the predefined maximum size $M$, all nodes in the component are annotated with a new pseudo label, and the corresponding pivot-neighbor pairs are also removed from set $\mathcal{E}$. The threshold $\tau_p$ is increased at the end of every iteration.

The process mentioned above is iterated until the pair set $\mathcal{E}$ is empty. The neglected orphan nodes are also annotated with new pseudo labels individually. Finally, the face nodes with same pseudo label constitute a cluster. To summary, Algorithm 1 lists the whole procedure of this label propagation. The maximum computational complexity of one iteration is of order $\mathcal{O}(3NK)$, and decreases as the iteration progresses. It is observed that involving face quality can help improve the clustering precision along with a slight drop of recall.

## 3 Experiments

### 3.1 Experimental Settings

**Datasets.** We use the IJB-C dataset [11] of low face quality as well as the refined MS1M dataset [4,7] of good face quality for training and testing in *face clustering*. The IJB-C contains about 138K face images from 3.5K identities, and the MS1M contains about 5.8M face images from 85K identities. The IJB-C dataset is randomly partitioned into 10 splits with equal identity number, and each part has the same distribution of nodes per identity. As IJB-C dataset is small, 9 parts are used for training to alleviate overfitting and 1 part for testing. The MS1M dataset is partitioned in the same way with 1 part for training and the other 9 parts for testing.

We evaluate the performance of face clustering by three commonly used metrics, i.e., *Pairwise* [16], *BCubed* [1] and *NMI* [19]. *Pairwise* and *BCubed* both measure the precision and recall of clustering, with F-score being their harmonic mean. The former metric emphasizes more on large clusters. *NMI* measures the global closeness of the output pseudo labels and the ground-truth. All the metrics have a range of $[0, 1]$ with 1 being the perfect. Four state-of-the-art algorithms, namely, L-GCN [20], GCN-DS [23], GCN-VE [22], and STAR-FC [15] are used for comparison.

Our framework is implemented in Pytorch. We first use the pre-trained IResNet50 [4,5] to extract the identity features of face nodes with dimension $d_f = 512$, and EQFace [9] to obtain the corresponding face quality scores. All the competing algorithms share the same input. We then set the neighbor number $K = 80$, the dimension of *key*, *query* and *value* $d_s = d_v = 2048$, and the number of multi-head attention $H = 2$. We also empirically set the initial linkage threshold $\tau_p = 0.9$, the quality threshold $\tau_q = 0.3$, and the maximum cluster size $M = 1000$. The maximum iteration number $T$ is set to 20 which is sufficient to obtain a satisfactory clustering. These parameter values remain the same in following experiments.

In addition, we train the model with 20 epochs from scratch, and optimize the loss with the SGD optimizer. The weight decay and the momentum are set to 0.0005 and 0.9, respectively. The initial learning rate is set to 0.01 and is empirically divided by 10 at 8, 12 and 18 epochs. All the experiments are performed on a single Tesla-P40 GPU, and one can use more for acceleration.

**Table 1.** Results on IJB-C dataset and MS1M dataset. "P" is short for "*Pairwise*", and "B" is short for "*BCubed*"

| | IJB-C | | | MS1M | | |
|---|---|---|---|---|---|---|
| | P F-score | B F-score | NMI | P F-score | B F-score | NMI |
| L-GCN | 0.876 | 0.818 | 0.925 | 0.959 | 0.975 | 0.994 |
| GCN-D | 0.686 | 0.710 | 0.871 | 0.899 | 0.906 | 0.980 |
| GCN-DS | 0.614 | 0.649 | 0.869 | 0.857 | 0.880 | 0.975 |
| GCN-V | 0.585 | 0.526 | 0.863 | 0.961 | 0.927 | 0.981 |
| GCN-VE | 0.535 | 0.474 | 0.844 | 0.975 | 0.963 | 0.991 |
| STAR-FC | 0.814 | 0.826 | 0.931 | **0.989** | 0.981 | 0.995 |
| FC-Q | **0.917** | **0.856** | **0.941** | 0.987 | **0.982** | **0.995** |

### 3.2 Experimental Results

Table 1 first shows the competing results on IJB-C dataset. The GCN-D only uses its detection module. The GCN-DS further incorporates the segmentation module but archives no performance improvement. This is because the algorithm fails in learning the complex cluster patterns as the identity similarities are not so reliable among low-quality faces. Algorithms GCN-V and GCN-VE present the same phenomenon, where the corresponding vertex confidence and edge connectivity estimation modules are heavily dependent on the identity similarity in feature subspace. The STAR-FC performs relatively better under the influence of its structure-preserved sub-graph sampling strategy. Overall, our FC-Q algorithm outperforms other algorithms on this dataset, and achieves 91.7% *pairwise* F-score under the employment of face quality scores. Table 1 also shows the competing results on MS1M dataset. It is observed that all the algorithms achieve performance improvements when the faces are in good quality. Overall, although the assistance role of face quality is diluted, our FC-Q algorithm still gets the satisfactory result. This indicates that our algorithm can be the first choice when implementing face clustering under general circumstance where the face quality is unknown or mixed.

### 3.3 Ablation Studies

In this subsection, we evaluate some design elements used in our algorithm. The following experiments are all conducted on IJB-C dataset. Table 2 presents the clustering results of our algorithm when individually removing the following four design elements, i.e., concatenating the quality scores, concatenating the Pearson correlation encoding, adding the prior relevancy matrix, and making *key* and *query* share the same projection. It is observed that all these four design elements contribute to the performance improvement, and the last element helps the most.

**Table 2.** Results on IJB-C when four design elements are removed successively. "P" is short for "*Pairwise*", and "B" is short for "*BCubed*".

| | P F-score | B F-score | NMI |
|---|---|---|---|
| FC-Q | **0.917** | **0.856** | **0.941** |
| Remove $q$ in (1) | 0.902 | 0.848 | 0.938 |
| Remove $\mathbf{p}$ in (1) | 0.895 | 0.799 | 0.921 |
| Remove $\mathbf{A}$ in (4) | 0.886 | 0.840 | 0.935 |
| Remove *Sharing* $\mathbf{W}^S$ in (3) | 0.828 | 0.726 | 0.883 |

**Table 3.** Results of L-GCN and FC-Q algorithms with their label propagation modules swapped. "A+B" denotes using A for linkage prediction and B for label propagation.

| | Pairwise | | | Bcubed | | |
|---|---|---|---|---|---|---|
| | Pre | Rec | F-score | Pre | Rec | F-score |
| L-GCN+L-GCN | 0.869 | 0.882 | 0.876 | 0.890 | 0.756 | 0.818 |
| L-GCN+FC-Q | 0.935 | 0.870 | 0.901 | 0.942 | 0.730 | 0.823 |
| FC-Q+L-GCN | 0.943 | 0.885 | 0.913 | 0.961 | 0.771 | 0.855 |
| FC-Q+FC-Q | **0.952** | **0.885** | **0.917** | **0.964** | **0.770** | **0.856** |

We further swap the label propagation modules of L-GCN and FC-Q to validate the effectiveness of incorporating the face quality. We can see from Table 3 that equipping our proposed label propagation module evidently improves the clustering precision of L-GCN with a slight drop of recall. This is because our module can further suppress the abnormal pairing based on extra local quality information. The improvement is not so significant on FC-Q as its linkage prediction module has already incorporated the face quality to output more reliable linkage likelihood.

Figure 4 further shows the effectiveness of our modified self-attention mechanism. An attention map extracted from the last layer is list on the left, which involves one pivot example and its $K = 80$ nearest neighbors. The neighbors are sorted in descend order with respect to the similarity. The attention values are taken as logarithm to better demonstrate the numerical differences. The sequence above the attention map describes the identity consistency, where the black dot indicates the neighbor node at same sequence position having the different identity. Three pivot-neighbor pairs are also list on the right, with quality score shown on the image and similarity below the arrow. We can see that the first pivot-neighbor pair in red box and the second pivot-neighbor pair in rosy box have close similarity but opposite identity consistency. The self-attention mechanism successfully suppresses the attention values of the former neighbor node based on the large score difference prior. Note that although the third neighbor node in black box with different identity has small score difference, the self-attention mechanism can also suppress its attention value based on the relevancy aggregation among involved nodes.

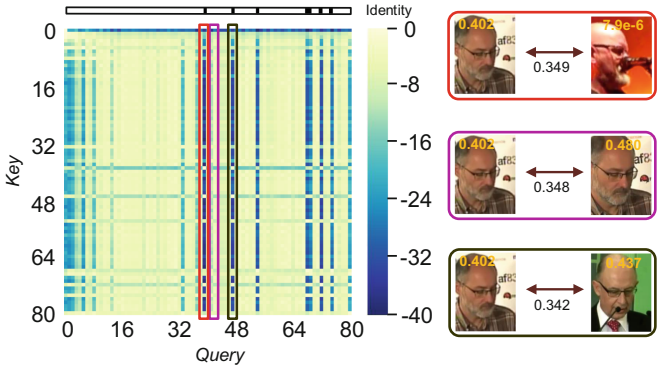


**Fig. 4.** Effect of self-attention mechanism. One attention map is list on the right and three pivot-neighbor pairs are list on the right.

## 4 Conclusion

This paper has introduced a face clustering algorithm, referred as FC-Q, to tackle with face nodes with mixed quality. The algorithm takes face quality score as extra input, which is incorporated into the linkage prediction module and label propagation module as a prior. The algorithm first modifies the Transformer encoder, and uses quality relevancy to infer more reliable linkage likelihood. Then the algorithm utilizes the local quality information to further suppress the abnormal pairing with high linkage likelihood. Experimental results validate that our algorithm gets the satisfactory clustering result under general circumstance where the face quality is unknown or mixed.

## References

1. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. Inf. Retriev. **12**(4), 461–486 (2009)
2. Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson Correlation Coefficient, pp. 1–4. Springer, Vienna (2009)
3. Chen, D., Lv, J., Zhang, Y.: Unsupervised multi-manifold clustering by learning deep representation. In: The Thirty-First AAAI Conference on Artificial Intelligence (2017)
4. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4690–4699 (2019)
5. Duta, I.C., Liu, L., Zhu, F., Shao, L.: Improved residual networks for image and video recognition. arXiv preprint arXiv:2004.04989 (2020)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A a density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)

7. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: a dataset and benchmark for large-scale face recognition. In: European Conference on Computer Vision, pp. 87–102 (2016)
8. Li, P., Zhao, H., Liu, H.: Deep fair clustering for visual learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9070–9079 (2020)
9. Liu, R., Tan, W.: Eqface: a simple explicit quality network for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1482–1490 (2021)
10. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
11. Maze, B., et al.: Iarpa janus benchmark - c: face dataset and protocol. In: 2018 International Conference on Biometrics, pp. 158–165 (2018)
12. Meng, Q., Zhao, S., Huang, Z., Zhou, F.: Magface: a universal representation for face recognition and quality assessment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 14225–14234 (2021)
13. Peng, B., Liu, M., Yang, H., Zhang, Z., Li, D.: Deep tiny network for recognition-oriented face image quality assessment. arXiv preprint arXiv:2004.04989 (2021)
14. Peng, X., Feng, J., Zhou, J.T., Lei, Y., Yan, S.: Deep subspace clustering. IEEE Trans. Neural Netw. **31**(12), 5509–5521 (2020)
15. Shen, S., et al.: Structure-aware face clustering on a large-scale graph with $\mathbf{10}^{\mathbf{7}}$ nodes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9085–9094 (2021)
16. Shi, Y., Otto, C., Jain, A.K.: Face clustering: Representation and pairwise constraints. IEEE Trans. Inf. Forens. Secur. **13**(7), 1626–1640 (2018)
17. Terhorst, P., Kolf, J.N., Damer, N., Kirchbuchner, F., Kuijper, A.: Ser-fiq: unsupervised estimation of face image quality based on stochastic embedding robustness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5651–5660 (2020)
18. Vaswani, A., et al.: Attention is all you need. In: International Conference on Neural Information Processing Systems, vol. 30, pp. 5998–6008 (2017)
19. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**(95), 2837–2854 (2010)
20. Wang, Z., Zheng, L., Li, Y., Wang, S.: Linkage based face clustering via graph convolution network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1117–1125 (2019)
21. Xiong, R., et al.: On layer normalization in the transformer architecture. In: International Conference on Machine Learning, pp. 10524–10533 (2020)
22. Yang, L., Chen, D., Zhan, X., Zhao, R., Loy, C.C., Lin, D.: Learning to cluster faces via confidence and connectivity estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 13369–13378 (2020)
23. Yang, L., Zhan, X., Chen, D., Yan, J., Loy, C.C., Lin, D.: Learning to cluster faces on an affinity graph. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2298–2306 (2019)