FC-DGCN: Deep Graph Convolutional Network for Face Clustering and Recognition

Ling Ding[1,2], Di Jin[1,3], Zhizhi Yu[1,4], Liang Yang[5], Dongxiao He[1,3]*

[1]College of Intelligence and Computing, Tianjin University
[2]School of Artificial Intelligence, Tianjin University
[3]School of Computer Science and Technology, Tianjin University
[4]School of Computer Software, Tianjin University
[5]School of Artificial Intelligence, Hebei University of Technology
{dltjdx2022, jindi, yuzhizhi, hedongxiao}@tju.edu.cn
yangliang@vip.qq.com

Abstract

Face recognition has achieved strong performance, but further gains often require much larger labeled datasets, making annotation costly. This motivates the use of unlabeled data, where face clustering is important for applications such as annotation and recognition. Recent GCN-based methods have shown promising results on affinity graphs, but balancing clustering quality and scalability on large-scale data remains challenging. To solve this problem, we propose FC-DGCN, a face clustering framework based on deep residual GCNs. The proposed method formulates clustering on an affinity graph through two complementary tasks: node confidence estimation and edge connectivity estimation. Specifically, we develop DGCN-V to estimate node confidence on the global graph and DGCN-E to estimate edge connectivity on local subgraphs. To better characterize local cluster structure, we further revise the confidence estimation target. Based on these two modules, FC-DGCN links each node to a higher-confidence neighbor with strong predicted connectivity, which naturally induces identity-consistent clusters. Experiments on MS-Celeb-1M show that FC-DGCN outperforms competitive baselines in Pairwise F-score and BCubed F-score while remaining computationally practical. Moreover, the clusters generated by FC-DGCN provide effective pseudo-labels for low-label face recognition, leading to substantial improvements on MegaFace and IJB-A.

1 Introduction

Deep neural networks trained on large-scale annotated datasets have driven remarkable progress in face recognition. However, further improvements increasingly rely on ever larger amounts of labeled data, making annotation prohibitively expensive in practice [1, 2]. Meanwhile, vast collections of unlabeled face images are readily available. Exploiting their underlying structure to derive supervisory signals offers a promising way to reduce dependence on manual annotation [3, 4]. In this context, face clustering is an important problem because it can organize unlabeled face images by identity [5, 6, 7, 8, 9] and provide useful supervision for downstream applications such as annotation and recognition [10, 11].

Traditional clustering methods such as K-means [12] and DBSCAN [13] often struggle on large-scale face data. Face representations are typically high-dimensional, and simple distance-based criteria are often insufficient to capture identity-level similarity reliably [14]. In addition, the scale of modern face

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

datasets poses substantial computational challenges for conventional clustering algorithms. Recent progress in deep clustering has substantially improved representation learning for complex data distributions [15, 16, 17, 18]. Graph-based representations provide a natural mechanism for modeling pairwise relations and neighborhood structure [19, 20]. This makes them particularly appealing for face clustering, where relational information among face images can guide identity-consistent grouping [21, 22, 23, 24].

In particular, recent Graph Convolutional Network-based (GCN-based) methods have shown that face clustering can be formulated as structured reasoning on an affinity graph [21, 24]. A representative framework decomposes the task into node confidence estimation and edge connectivity estimation, which together provide an effective mechanism for building identity-consistent clusters [21]. However, within this line of work, graph reasoning modules still have limited expressive capacity, and the trade-off between clustering quality and scalability remains challenging on large datasets [21, 24]. Moreover, although deep residual graph models have been explored for local link prediction in face clustering, their integration into a confidence-and-connectivity framework for large-scale affinity-graph clustering remains underexplored.

To address these issues, we propose **FC-DGCN**, a face clustering approach based on deep residual GCNs. We model each face image as a node in an affinity graph and decompose clustering into two complementary sub-tasks: node confidence estimation and edge connectivity estimation [21]. The first sub-task predicts how confidently a node lies in the interior of a cluster, while the second estimates whether two neighboring nodes belong to the same identity. Based on these two signals, each node is linked to a higher-confidence neighbor with strong connectivity, which naturally induces cluster structures on the graph. To realize this idea, we design two learnable modules, namely a confidence estimation module and a connectivity estimation module, both built on deep residual GCNs with shortcut connections [25, 26]. Compared with prior confidence-and-connectivity frameworks, our method strengthens graph reasoning by using deeper residual graph propagation and a revised confidence estimation design [21]. This enables the model to aggregate richer structural context while maintaining stable optimization in deeper graph networks [26]. Experimental results show that our model achieves stronger clustering performance than competitive baselines in terms of Pairwise F-score and BCubed F-score, while remaining computationally practical on large-scale data. Furthermore, the clusters produced by FC-DGCN provide effective pseudo-labels for low-label face recognition and substantially improve recognition accuracy over the initial supervised models [1, 2].

The main contributions of this work are threefold.

- We propose FC-DGCN, a face clustering approach based on deep residual graph convolutional networks for affinity-graph modeling.
- We instantiate FC-DGCN with two learnable modules for node confidence estimation and edge connectivity estimation, enabling more expressive graph reasoning for face clustering.
- We show that the resulting clusters can provide effective pseudo-labels for face recognition under limited-label settings, leading to substantial gains over the initial recognition models.

2 Preliminaries

We first give the affinity-graph formulation, then give the confidence and connectivity paradigm.

Affinity-Graph Formulation. Given a face dataset with N images, first extract a D-dimensional feature vector $\mathbf{h}_i \in \mathbb{R}^D$ for each image using a pretrained face encoder, and stack them into a feature matrix $H = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N]^\top \in \mathbb{R}^{N \times D}$. Based on these features, we construct a K nearest neighbor affinity graph $G = (V, E)$, where each node $v_i \in V$ corresponds to one face image, and edges connect visually similar neighboring samples [16, 17]. The affinity between two nodes v_i and v_j is measured by the cosine similarity between their feature vectors. The resulting graph is represented by an adjacency matrix $A \in \mathbb{R}^{N \times N}$. If nodes v_i and v_j are disconnected, then $a_{i,j} = 0$.

Confidence and Connectivity Paradigm. Following prior graph-based face clustering formulations, face clustering can be cast as a structured reasoning problem on the affinity graph [17, 27]. Instead of directly partitioning the graph in a single step, the task is decomposed into two complementary sub-tasks: node confidence estimation and edge connectivity estimation [5, 27]. The confidence of a node reflects the degree of local cluster consistency around that node, and nodes with high confidence are more likely to lie in the interior of coherent identity clusters [5, 21, 14]. In contrast, edge connectivity evaluates whether a pair of neighboring nodes should be linked, namely, whether they are likely to belong to the same identity.

Based on these two quantities, clustering is performed by linking each node to a neighboring node with higher confidence and strong predicted connectivity [21, 28]. Repeating this local linking process over the affinity graph naturally induces a set of tree-like cluster structures, from which final face clusters can be obtained [17]. This paradigm provides the basis for our method: node confidence offers a global structural cue for identifying reliable cluster interiors, while edge connectivity provides a local discriminative cue for selecting valid cross-node links.

3 Methodology

We first give a brief overview, and then introduce the proposed new methods in detail.

3.1 Overview

Building on the affinity-graph formulation and the confidence-and-connectivity paradigm introduced in Section 2, we propose FC-DGCN, a deep residual graph approach for face clustering. The main idea is to enhance graph reasoning on large and complex affinity graphs by replacing the shallow GCN modules in prior methods with deeper residual graph networks, thereby better capturing multi-hop structural context and improving clustering performance. As illustrated in Figure 1, FC-DGCN consists of two synergistic components: (1) **DGCN-V**, a deep residual graph network for node confidence estimation over the global affinity graph, and (2) **DGCN-E**, a deep residual graph convolutional network for edge connectivity estimation on local subgraphs. In addition, we revise the confidence estimation target to better characterize local cluster consistency. Specifically, we define the confidence target using the average affinity to same-identity neighbors in the local neighborhood, which provides a simpler and more direct measure of local intra-cluster coherence. Based on these two modules, the proposed approach performs face clustering by jointly modeling reliable cluster interiors and discriminative inter-node connections.

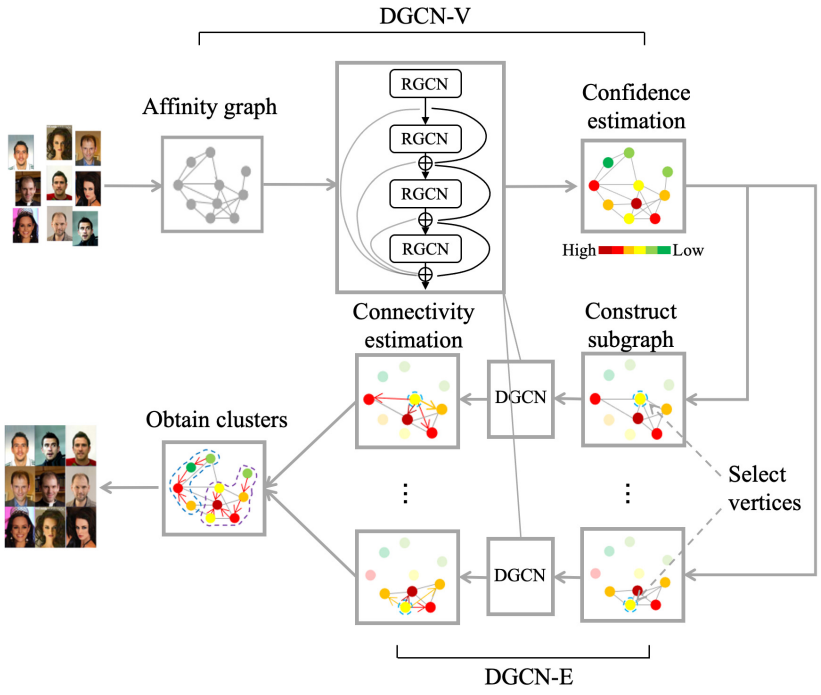


Figure 1: The overall approach of FC-DGCN.

3.2 Confidence Estimation Module

The confidence estimation module aims to predict a confidence score for each node on the global affinity graph, indicating how reliably the node lies in a locally coherent cluster region. Intuitively, nodes surrounded by neighbors from the same identity tend to have high confidence, while nodes near cluster boundaries or in structurally ambiguous regions tend to have low confidence. Inspired

by Yang et al. [21], and to model such structural patterns, we employ a deep residual GCN, termed **DGCN-V**, which takes the graph structure and node features as input and outputs the confidence prediction for each node. Formally, the confidence target for node v_i is defined as:

$$c_i = \frac{1}{|\mathcal{N}_i|} \sum_{v_j \in \mathcal{N}_i} \mathbb{I}[y_j = y_i] \cdot A_{ij}, \tag{1}$$

where $\mathcal{N}_i$ denotes the neighbor set of node v_i, y_i is the ground-truth identity label of v_i, and A_{ij} is the affinity between nodes v_i and v_j. Based on this target, DGCN-V learns to regress node-wise confidence scores over the whole graph. Concretely, DGCN-V takes the adjacency matrix A and the node feature matrix $H = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N]^\top \in \mathbb{R}^{N \times D}$ as input. Let $X^{(0)} = H$ denote the input node embeddings. The propagation rule at the l-th layer is defined as:

$$X^{(l+1)} = \sigma\left(g\left(\tilde{A}, X^{(l)}\right) W^{(l)}\right), \tag{2}$$

where $\tilde{A} = \hat{D}^{-1}(A + I)$, $\hat{D}_{ii} = \sum_j (A + I)_{ij}$, $W^{(l)}$ is a trainable weight matrix, and $\sigma(\cdot)$ denotes the ReLU activation function [29]. The operator $g(\tilde{A}, X^{(l)})$ concatenates the current node embedding and its neighborhood-aggregated embedding, for example:

$$g\left(\tilde{A}, X^{(l)}\right) = \left[X^{(l)},\ \tilde{A}X^{(l)}\right]. \tag{3}$$

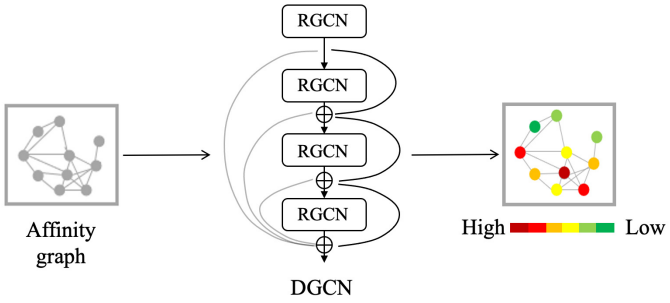


Figure 2: The overview of the DGCN-V module.

To enable deeper graph reasoning, we introduce residual graph convolutional blocks in DGCN-V. As illustrated in Figure 2, each residual block contains four graph convolution layers together with a shortcut connection, which helps preserve input information and facilitates optimization in deep architectures [25, 26]. The residual update is formulated as:

$$X^{(l+1)} = X^{(l)} + \phi\big(X^{(l)}; \Theta^{(l)}\big), \tag{4}$$

where $X^{(l)}$ and $X^{(l+1)}$ denote the input and output embeddings of the l-th residual block, respectively, and $\phi(\cdot; \Theta^{(l)})$ denotes the residual mapping parameterized by $\Theta^{(l)}$.

Stacking multiple residual blocks yields the final node embeddings $X^{(L)}$, which encode richer multi-hop structural information on the affinity graph. Based on the final-layer embeddings, the predicted confidence scores are obtained through a linear regressor:

$$\hat{\mathbf{c}} = X^{(L)} W_c + b_c, \tag{5}$$

where $\hat{\mathbf{c}} \in \mathbb{R}^N$ is the predicted confidence vector, W_c is the regression weight matrix, and b_c is the bias term. Accordingly, the predicted confidence of node v_i is given by the i-th element $\hat{c}_i$ of $\hat{\mathbf{c}}$.

Given the ground-truth confidence target defined in Eq. 1, DGCN-V is trained by minimizing the mean squared error between the predicted and target confidence values:

$$\mathcal{L}_V = \frac{1}{N} \sum_{i=1}^{N} (c_i - \hat{c}_i)^2. \tag{6}$$

By optimizing this objective, DGCN-V learns node representations that capture local structural consistency and produces confidence estimates for subsequent candidate selection.

3.3 Connectivity Estimation Module

The connectivity estimation module aims to predict whether two neighboring nodes should be linked for clustering. For a node v_i, neighbors with higher confidence are more likely to lie in the interior of a coherent cluster. However, higher confidence alone does not guarantee identity consistency. We therefore introduce a second module, termed **DGCN-E** (as shown in Figure 3), to estimate the connectivity between v_i and its candidate neighbors based on local subgraph structure.

Given the confidence predictions from DGCN-V, we construct a candidate set for each node v_i as:

$$\mathcal{S}_i = \{v_j \mid \hat{c}_j > \hat{c}_i,\ v_j \in \mathcal{N}_i\}\,. \tag{7}$$

The candidate set contains neighboring nodes with higher confidence than v_i, which serve as potential linking targets during cluster construction. For each candidate pair (v_i, v_j) with $v_j \in \mathcal{S}_i$, DGCN-E operates on the corresponding local subgraph and predicts their connectivity score.

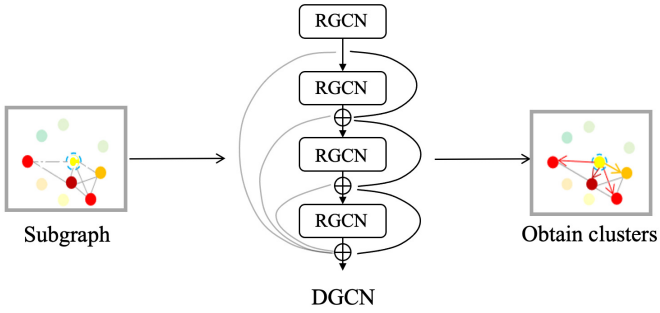


Figure 3: The overview of the DGCN-E module.

Given ground-truth identity labels during training, the supervision target is defined as:

$$r_{ij} = \begin{cases} 1, y_i = y_j, \\ 0, y_i \neq y_j, \end{cases} \qquad v_j \in \mathcal{S}_i\ , \tag{8}$$

where r_{ij} indicates whether nodes v_i and v_j belong to the same identity. The objective of DGCN-E is to learn a mapping from the local subgraph representation to the predicted connectivity score $\hat{r}_{ij}$.

Similar to DGCN-V, DGCN-E adopts a deep residual graph convolutional architecture with shortcut connections to enhance structural reasoning on graph data. Let $\mathcal{G}_{ij}$ denote the local subgraph associated with the candidate pair (v_i, v_j). DGCN-E encodes $\mathcal{G}_{ij}$ into a pairwise representation and predicts:

$$\hat{r}_{ij} = f_{\mathrm{DGCN-E}}(\mathcal{G}_{ij}), \tag{9}$$

where $f_{\mathrm{DGCN-E}}(\cdot)$ denotes the connectivity estimation network. The module is trained by minimizing the mean squared error over all candidate pairs:

$$\mathcal{L}_E = \frac{1}{\sum_{i=1}^{N} |\mathcal{S}_i|} \sum_{i=1}^{N} \sum_{v_j \in \mathcal{S}_i} (r_{ij} - \hat{r}_{ij})^2\,. \tag{10}$$

At inference time, each node is linked to a neighboring node with higher predicted confidence and high connectivity score, and the resulting directed links induce tree-like cluster structures on the affinity graph.

3.4 Face Recognition with Pseudo Labels

Face recognition performance generally benefits from large-scale annotated datasets, but collecting identity labels at such scale is costly and often impractical. To alleviate this issue, we build a face recognition approach on top of FC-DGCN, where the clustering results on unlabeled face images are used as pseudo labels to augment supervised training.

As illustrated in Figure 4, let the face dataset be divided into a labeled subset $\mathcal{D}_L$ and an unlabeled subset $\mathcal{D}_U$. We first train an initial face recognition model M_0 on $\mathcal{D}_L$ using the available ground-truth

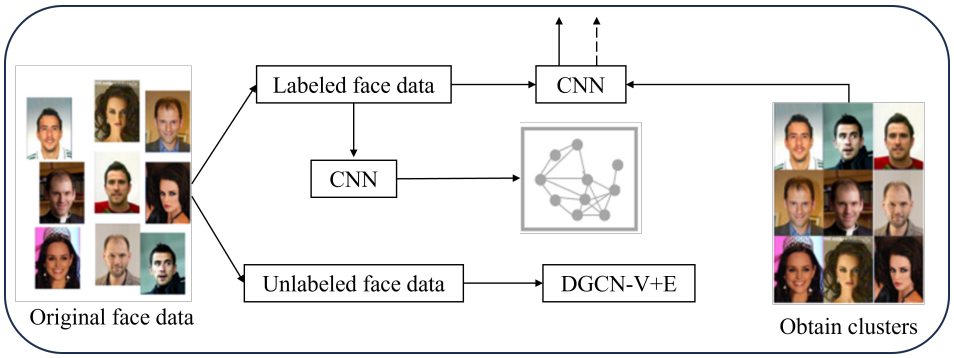


Figure 4: Framework of the face recognition model based on FC-DGCN.

identity annotations. The trained model is then used to extract face embeddings for samples in $\mathcal{D}_U$, from which we construct the affinity graph and apply FC-DGCN to obtain clustering results. The cluster assignments produced by FC-DGCN are treated as pseudo labels for unlabeled samples, yielding a pseudo-labeled set $\mathcal{D}_P$. Finally, we train the face recognition model on the union of the labeled and pseudo-labeled data $\mathcal{D}_{\text{train}} = \mathcal{D}_L \cup \mathcal{D}_P$, so that the model can benefit from both reliable manual annotations and large-scale pseudo supervision. In this way, FC-DGCN transfers structural clustering knowledge from unlabeled data into the recognition stage, improving face recognition performance when labeled data are limited.

4 Experiments

We first present the experimental setup, including the datasets, evaluation metrics, and baselines. Then, we compare FC-DGCN with SOTA methods on the face clustering task. Next, we evaluate the effectiveness of the proposed pseudo-label-based face recognition framework on public face recognition benchmarks under limited labeled data. Finally, we conduct ablation studies to analyze the contributions of different initial recognition models and the parameter ρ.

4.1 Experimental Setup

Datasets. We use MS-Celeb-1M [30] for face clustering and pseudo-label generation, and evaluate face recognition performance on MegaFace [31] and IJB-A [32]. In our experiments, we use a subset of MS-Celeb-1M containing 86K identities and 5.8 million images [2, 33]. MegaFace is adopted for large-scale face identification evaluation, while IJB-A is used for face verification evaluation [2, 23]. Example face images from MS-Celeb-1M are shown in Figure 5. Details are in Appendix B.



Figure 5: Samples of the training images of MS-Celeb-1M dataset.

Evaluation metrics. For face clustering, we adopt **Pairwise F-score** and **BCubed F-score**, denoted by F_P and F_B [34]. For face recognition, we follow the standard evaluation protocols of MegaFace and IJB-A, and report top-1 accuracy and verification rate [2, 23].

Baselines. We compare FC-DGCN with a range of clustering methods, including K-means [12], DBSCAN [13], CDP [2], L-GCN [22], LTC [23], GCN-V+E [21]. Detailed implementation settings are provided in Appendix A.

4.2 Face Clustering Results

We evaluate the proposed method on the face clustering task using the MS-Celeb-1M [30] dataset. The quantitative comparison is reported in Table 1. The results show that traditional clustering methods exhibit clear limitations on large-scale face data. Although K-means [12] achieves relatively competitive F-scores when the number of clusters is known, it is prohibitively slow in practice. DBSCAN [13] is highly efficient, but its clustering accuracy is substantially lower. In contrast, graph-based deep clustering methods provide a better trade-off between efficiency and accuracy.

Table 1: Comparison of performance of face clustering.

Method	F_P	F_B	TIME
K-means [12]	79.23	81.01	13.7h
DBSCAN [13]	67.72	67.24	3.5m
CDP [2]	75.02	78.70	**2.8m**
L-GCN [22]	78.53	84.36	92.6m
LTC [23]	85.59	85.42	71.3m
GCN-V+E [21]	87.55	85.94	11.6m
FC-DGCN	**90.23**	**88.49**	15.8m

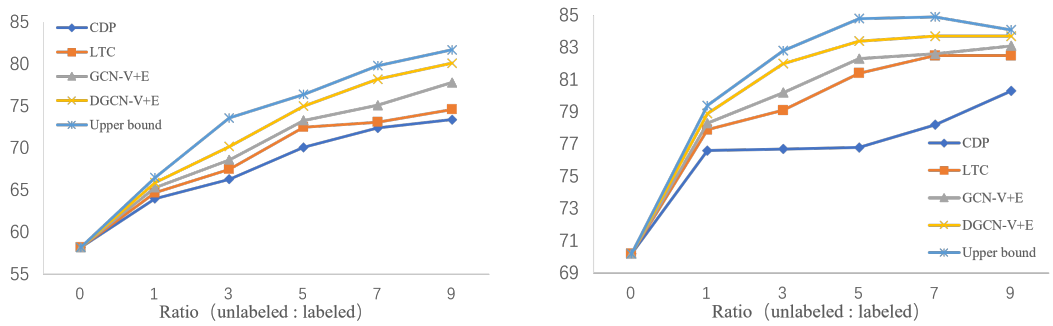
Among all compared methods, FC-DGCN achieves the best overall performance, improving both F_P and F_B over previous GCN-based methods. Compared with GCN-V+E, the proposed method yields clear gains in clustering quality, which verifies the advantage of deeper residual graph reasoning. Although FC-DGCN is slightly slower than GCN-V+E [21] due to its deeper architecture, it remains significantly more efficient than several other graph-based baselines such as L-GCN [22] and LTC [23]. In practical applications, the DGCN-E module can also be parallelized over different nodes to further improve scalability.

4.3 Face Recognition with Pseudo Labels

We next evaluate the proposed pseudo-label-based face recognition framework on MS-Celeb-1M [30], MegaFace [31], and IJB-A [32]. Following the setting described in Section 3, we divide the training data into labeled and unlabeled subsets. Specifically, the MS-Celeb-1M subset is split into 10 equal parts with similar identity distributions. One part is randomly selected as the labeled subset, denoted by S_T, while the remaining nine parts are treated as the unlabeled subset, denoted by S_U.

We first train an initial face recognition model M_L on S_T. The trained model is then used to extract face embeddings, based on which FC-DGCN is applied to cluster the unlabeled data and generate pseudo-labels. The resulting pseudo-labeled set is denoted by S_F. Finally, the face recognition model is trained on $S_T \cup S_F$ to obtain the final recognition model. S_T is used to train both the initial face recognition model and the clustering model. As an upper bound, we also report the performance of the same face recognition model trained with all manually labeled data, denoted by S_A.

As shown in Figure 6a, the horizontal axis represents the ratio of unlabeled data to labeled data, and the vertical axis represents top-1 accuracy on MegaFace. The proposed framework based on FC-DGCN improves the performance of the initial face recognition model from 58.21% to 80.32%. As shown in Figure 6b, the horizontal axis represents the ratio of unlabeled data to labeled data, and the vertical axis represents the verification rate on IJB-A. Compared with the initial face recognition model, the proposed framework improves the performance from 70.23% to 83.81%, and even surpasses the model trained using all manually labeled data, which may be attributed to the existence of labeling noise in the IJB-A dataset.



(a) Top-1 recognition rate on the MegaFace dataset.

(b) Top-1 recognition rate on the IJB-A dataset.

Figure 6: Face recognition performance under different unlabeled-to-labeled ratios.

4.4 Ablations

We conduct ablation studies to analyze the effectiveness of the proposed pseudo-label framework under different initial face recognition models, as well as the impact of the parameter ρ in the clustering stage.

Different initial face recognition models. In the main experiments, ResNet [26] is used as the initial face recognition model. To further verify the generality of the proposed framework, we additionally consider Inception [27] and CosFace [35] as alternative initial models. The results on MegaFace and IJB-A are summarized in Table 2. The results show that FC-DGCN consistently improves recognition performance for all initial models. On MegaFace, the performance improves by 21.91%, 21.54%, and 17.28% for ResNet, Inception, and CosFace, respectively. On IJB-A, the corresponding improvements are 13.58%, 14.85%, and 10.29%. Moreover, in several cases, training with pseudo labels not only substantially outperforms training with S_T alone, but also approaches or even exceeds the performance obtained using all manually labeled data.

Table 2: Performance comparison of different initial face recognition models on MegaFace and IJB-A.

Setting	MegaFace			IJB-A		
	ResNet	Inception	CosFace	ResNet	Inception	CosFace
S_T	58.21	59.82	76.49	70.23	72.45	86.45
$S_T + S_F$	80.32	81.36	93.77	**83.81**	**87.30**	**96.74**
S_A	**82.65**	**82.42**	**95.62**	82.16	85.62	96.01

Effect of ρ. We further study the effect of the parameter ρ, which controls the proportion of nodes processed by DGCN-E. We vary ρ from 0 to 1 with a step size of 0.1. As shown in Figure 7, the leftmost point ($\rho = 0$) corresponds to the result without the DGCN-E module, while the rightmost point ($\rho = 1$) corresponds to applying DGCN-E to all nodes. Focusing only on 10% of the nodes with high confidence ($\rho = 0.1$) already yields considerable performance gains while adding little computational cost. As ρ increases, more nodes benefit from the connectivity estimation of DGCN-E, which generally improves F_P in the early stage. However, when DGCN-E is applied to too many low-confidence nodes, the performance slightly drops. This is likely because the connectivity patterns around low-confidence nodes are more ambiguous, making them harder for graph neural networks to model reliably.

5 Related Work

Unsupervised Face Clustering. Face clustering aims to group face images by identity without requiring labels, and has long been studied as a fundamental problem in computer vision and pattern recognition. Early methods mainly relied on conventional clustering algorithms or hand-crafted similarity measures. For example, K-means [12] and DBSCAN [13] are representative general-

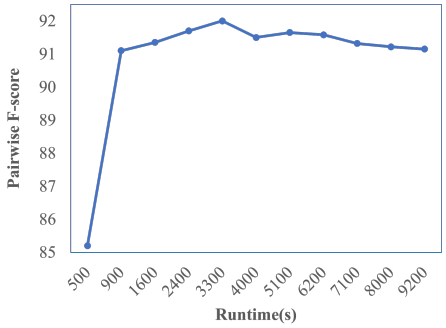


Figure 7: Ablation study of the parameter ρ.

purpose clustering methods, but they often struggle on large-scale face data due to high feature dimensionality, complex intra-class variation, and limited scalability.

To address these issues, prior face clustering works explored more specialized similarity modeling and clustering strategies. Otto et al. [5] proposed an approximate rank-order clustering method for clustering millions of faces, while Shi et al. [14] introduced pairwise constraints to improve face grouping. Lin et al. [36, 37] further developed density-aware and hierarchical methods for unconstrained face clustering, and Tapaswi et al. [33] studied face clustering when the number of clusters is unknown. These methods demonstrate the importance of robust similarity estimation and scalable clustering design, but they still rely heavily on manually designed criteria or limited local decision rules. Recent studies [11] also continue to revisit density-based formulations for face clustering, indicating that robust local structure modeling remains important in this problem.

Learning-based Face Clustering. With the development of deep representation learning, recent methods have increasingly formulated face clustering as a learning problem. Deep clustering methods have shown that learned representations can substantially improve clustering quality on complex visual data [15, 16, 17, 18]. In the face domain, several works incorporate learned affinities, graph structures, or pseudo-supervision to better exploit unlabeled data. Zhan et al. [2] proposed consensus-driven propagation to leverage relational information in massive unlabeled face data. Wang et al. [22] formulated face clustering as a linkage prediction problem and used GCNs to infer whether two samples should be connected. Yang et al. [23] further modeled face clustering on an affinity graph and demonstrated the effectiveness of graph-based structured reasoning. Zhao et al. [24] introduced a density-division strategy to reduce redundant computation by applying graph reasoning selectively to low-density regions. Compared with traditional unsupervised methods, these learning-based approaches better adapt to the structure of face data and improve both clustering quality and applicability to large-scale settings.

Deep and Scalable GCNs for Clustering. GCNs provide a natural way to model pairwise affinities and neighborhood structures in clustering problems. In face clustering, Yang et al. [21] proposed a confidence-and-connectivity framework that decomposes clustering into node confidence estimation and edge connectivity estimation, achieving strong performance on large-scale datasets. This framework offers an effective formulation for affinity-graph clustering and is closely related to our work. However, most existing GCN-based face clustering methods rely on relatively shallow graph reasoning modules, which may limit their ability to capture richer multi-hop structural context in large and complex affinity graphs [21, 22, 24]. Meanwhile, Qi et al. [25] explored a deep residual GCN for local link prediction in face clustering, showing the potential of deeper graph models with residual connections. Motivated by these advances, our work revisits the confidence-and-connectivity paradigm and develops a deeper residual graph framework for both node confidence estimation and edge connectivity estimation, aiming to improve graph expressiveness while remaining practical for large-scale face clustering.

6 Conclusions and Limitations

In this paper, we propose FC-DGCN, a face clustering framework based on deep graph convolutional networks with residual connections. By introducing a confidence estimation module and a connectivity

estimation module, the proposed method effectively models both node-level and edge-level relations on the affinity graph, leading to more accurate and efficient face clustering on large-scale datasets. Building on FC-DGCN, we further developed a pseudo-label-based face recognition framework for scenarios with limited annotated data. Experimental results show that the proposed method not only outperforms existing face clustering baselines in both clustering accuracy and efficiency, but also improves face recognition performance by leveraging large amounts of unlabeled face data.

Limitations. First, its performance relies on the quality of the initial feature embeddings used to build the KNN affinity graph. If the initial embeddings fail to preserve identity consistency, erroneous neighborhood relations may be introduced at the graph construction stage and further affect both confidence estimation and connectivity estimation. Second, the pseudo labels produced by FC-DGCN are not guaranteed to be error-free. In low-label settings, inaccurate cluster assignments may introduce noisy supervision into the subsequent recognition training stage, which can reduce or destabilize the gains from pseudo-label learning.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 92370111, No. 62272340, No. 62422210, No. 62276187, No. 62402337, No. 92570118 and No. 62376088), the Hebei Natural Science Foundation (No. F2024202047) and the Hebei Yanzhao Golden Platform Talent Gathering Programme Core Talent Project (Education Platform) (No. HJZD202509).

References

[1] Haiming Yu, Yin Fan, Keyu Chen, He Yan, Xiangju Lu, Junhui Liu, and Danming Xie. Unknown identity rejection loss: Utilizing unlabeled data for face recognition. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2662–2669, 2019.

[2] Xiaohang Zhan, Ziwei Liu, Junjie Yan, Dahua Lin, and Chen Change Loy. Consensus-driven propagation in massive unlabeled data for face recognition. In *ECCV*, pages 568–583, 2018.

[3] Kaishen Yuan, Zitong Yu, Xin Liu, Bohao Xing, Yuting Zhang, Weicheng Xie, Linlin Shen, and Björn W. Schuller. Multi-granularity facial emotional representation with unlabeled data and textual supervision. *IEEE Trans. Image Process.*, 35:3479–3494, 2026.

[4] Lin Li, Jun Xiao, Hanrong Shi, Hanwang Zhang, Yi Yang, Wei Liu, and Long Chen. Nicest: Noisy label correction and training for robust scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6873–6888, 2024.

[5] Charles Otto, Dayong Wang, and Anil K. Jain. Clustering millions of faces by identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):289–303, 2018.

[6] Yujiang Wang, Mingzhi Dong, Jie Shen, Yiming Luo, Yiming Lin, Pingchuan Ma, Stavros Petridis, and Maja Pantic. Self-supervised video-centralised transformer for video face clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12944–12959, 2023.

[7] Shuai Shen, Wanhua Li, Zheng Zhu, Jie Zhou, and Jiwen Lu. STAR-FC: structure-aware face clustering on ultra-large-scale graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):14005–14019, 2023.

[8] Dafeng Zhang, Yongqi Song, and Shizhuo Liu. Enhancing noise resilience in face clustering via sparse differential transformer. In *AAAI*, pages 12403–12411, 2026.

[9] Shuai Shen, Wanhua Li, Xiaobing Wang, Dafeng Zhang, Zhezhu Jin, Jie Zhou, and Jiwen Lu. Clip-cluster: Clip-guided attribute hallucination for face clustering. In *ICCV*, pages 20729–20738, 2023.

[10] Jun Dan, Yang Liu, Baigui Sun, Jiankang Deng, and Shan Luo. Transface++: Rethinking the face recognition paradigm with a focus on accuracy, efficiency, and security. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(2):1243–1261, 2026.

[11] Yu Zhou, Jiaoyang Cheng, Jianqiao Long, Jiguang Li, Jiaqing Li, and Jichun Li. Face clustering using a novel density peaks clustering algorithm. *Neurocomputing*, page 131576, 2025.

[12] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[13] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

[14] Yichun Shi, Charles Otto, and Anil K Jain. Face clustering: representation and pairwise constraints. *IEEE Transactions on Information Forensics and Security*, 13(7):1626–1640, 2018.

[15] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018.

[16] Paras Dahal. Learning embedding space for clustering from deep representations. In *Big data*, pages 3747–3755, 2018.

[17] Dongdong Chen, Jiancheng Lv, and Yi Zhang. Unsupervised multi-manifold clustering by learning deep representation. In *AAAI workshops*, 2017.

[18] Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, Philip S. Yu, and Lifang He. Deep clustering: A comprehensive survey. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):5858–5878, 2025.

[19] Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S. Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey. *IEEE Trans. Knowl. Data Eng.*, 38(7):4385–4404, 2026.

[20] Ling Ding, Chao Li, Di Jin, and Shifei Ding. Survey of spectral clustering based on graph theory. *Pattern Recognit.*, 151:110366, 2024.

[21] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. Learning to cluster faces via confidence and connectivity estimation. In *CVPR*, pages 13369–13378, 2020.

[22] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *CVPR*, pages 1117–1125, 2019.

[23] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *CVPR*, pages 2298–2306, 2019.

[24] Qingchao Zhao, Long Li, Yan Chu, Zhengkui Wang, and Wen Shan. Density division face clustering based on graph convolutional networks. In *ICPR*, pages 5017–5023, 2022.

[25] Chao Qi, Jianming Zhang, Hongjie Jia, Qirong Mao, Liangjun Wang, and Heping Song. Deep face clustering using residual graph convolutional network. *Knowledge-Based Systems*, 211:106561, 2021.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[28] Hong Zhu, Chuan Chen, Li-Zhi Liao, and Michael K Ng. Multiple graphs clustering by gradient flow method. *Journal of the Franklin Institute*, 355(4):1819–1845, 2018.

[29] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814, 2010.

[30] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. *arXiv preprint arXiv:1607.08221*, 2016.

[31] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, pages 4873–4882, 2016.

[32] Brendan F Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, and Anil K Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *CVPR*, pages 1931–1939, 2015.

[33] Makarand Tapaswi, Marc T Law, and Sanja Fidler. Video face clustering with unknown number of clusters. In *ICCV*, pages 5027–5036, 2019.

[34] Maoshan Liu, Yan Wang, and Zhicheng Ji. Self-supervised convolutional subspace clustering network with the block diagonal regularizer. *Neural Processing Letters*, 53(6):3849–3875, 2021.

[35] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, pages 5265–5274, 2018.

[36] Wei-An Lin, Jun-Cheng Chen, Carlos D Castillo, and Rama Chellappa. Deep density clustering of unconstrained faces. In *CVPR*, pages 8128–8137, 2018.

[37] Wei-An Lin, Jun-Cheng Chen, and Rama Chellappa. A proximity-aware hierarchical clustering of faces. In *FG*, pages 294–301, 2017.

A Experimental Details

Data split. Following the standard setting in prior work [21], we use the cleaned MS-Celeb-1M subset containing 86K identities and 5.8M face images. The dataset is randomly divided into 10 parts with similar identity distributions. Each part contains approximately 8.6K identities and 580K images. In the pseudo-label-based face recognition experiments, one part is randomly selected as the labeled subset, while the remaining nine parts are treated as unlabeled data.

Affinity graph construction. For all clustering experiments, face embeddings are extracted by a trained face recognition CNN, and a KNN affinity graph is constructed based on cosine similarity between embeddings. We set $K = 80$ on MS-Celeb-1M. To avoid unreliable links, we discard edges with similarity lower than the threshold $\tau = 0.8$.

Network configuration. Since DGCN-V operates on the global affinity graph with millions of nodes, we adopt a relatively computation-aware architecture and use DGCN-8 with two residual graph convolution modules, where each module contains four graph convolutional layers. In contrast, DGCN-E operates on local subgraphs with at most K nodes, and we therefore use a deeper DGCN-16 with four residual graph convolution modules to enhance local structural reasoning.

Optimization. For model training, we use momentum SGD with an initial learning rate of 0.1 and a weight decay of 1×10^{-5}.

Runtime measurement. The running time is measured on an NVIDIA RTX 2080 GPU. On MS-Celeb-1M, the reported runtime is measured when the number of unlabeled samples is 583K.

Baseline details. We compare FC-DGCN with several representative face clustering methods, including traditional clustering methods, graph-based methods, and supervised clustering approaches.

- **K-means [12]**: A classical partition-based clustering algorithm that groups samples by minimizing the within-cluster Euclidean distance. Although it can achieve competitive results when the number of clusters is known, its applicability is limited in large-scale face clustering where the true number of identities is usually unavailable.
- **DBSCAN [13]**: A representative density-based clustering method that identifies clusters as dense regions in the feature space and treats low-density samples as noise. It does not require the number of clusters to be specified in advance, but its performance may degrade when different identity clusters exhibit varying local densities.
- **CDP [2]**: A graph-based clustering approach that exploits consensus-driven propagation to progressively construct clusters from reliable pairwise relations. It is computationally efficient on large-scale unlabeled face data and serves as a strong non-GCN baseline in our experiments.
- **L-GCN [22]**: A supervised face clustering method based on graph convolutional networks. It formulates face clustering as a linkage prediction problem and uses GCNs to infer whether neighboring samples or subgraphs should be connected. By leveraging graph context, it improves over conventional pairwise prediction methods, but its inference cost is relatively high on large-scale datasets.
- **LTC [23]**: A supervised graph-based face clustering framework on affinity graphs. It reformulates clustering as a two-stage localization problem and combines graph reasoning with detection and segmentation style modules to identify clusters. LTC achieves strong clustering accuracy, but its iterative proposal-based design introduces additional computational overhead.
- **GCN-V+E [21]**: A confidence-and-connectivity based face clustering framework that decomposes clustering into two sub-problems: node confidence estimation and edge connectivity estimation. It employs one GCN, denoted as GCN-V, to predict the confidence of each node on the global affinity graph, and another GCN, denoted as GCN-E, to estimate the connectivity between candidate node pairs on local subgraphs.

B Dataset Details

MS-Celeb-1M dataset. We use MS-Celeb-1M [30] as the main dataset for face clustering and pseudo-label generation. Following prior work, we adopt the cleaned subset containing approximately 86K

identities and 5.8M face images. In the clustering experiments, this dataset is used to evaluate the large-scale clustering performance of different methods. In the pseudo-label-based face recognition experiments, the same subset is randomly divided into 10 parts with similar identity distributions, where each part contains about 8.6K identities and 580K images. One part is used as the labeled subset for training the initial recognition model and the clustering model, while the remaining nine parts are treated as unlabeled data for pseudo-label generation.

MegaFace dataset. We use MegaFace [31] to evaluate large-scale face recognition performance. Following the standard identification protocol, we report the top-1 identification accuracy on MegaFace. In our framework, MegaFace is only used for evaluation and is not involved in the training or pseudo-label generation process.



Figure 8: Samples of the training images of the MegaFace dataset.

IJB-A dataset. We use IJB-A [32] as an additional benchmark for face verification evaluation. Following standard practice, we report the verification rate on IJB-A. Similar to MegaFace, IJB-A is only used for evaluation and does not participate in training.



Figure 9: Samples of the training images of the IJB-A dataset.

The role of each dataset in the proposed framework is summarized as follows: MS-Celeb-1M is used for face clustering and pseudo-label generation, while MegaFace and IJB-A are used to evaluate the recognition models trained with labeled and pseudo-labeled data.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the contributions of our paper, including the motivation and design of the proposed FC-DGCN.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations in conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All theoretical results are accompanied by clearly stated assumptions and complete proofs, provided in the main paper and referenced appropriately.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The supplemental material contains a file of our model's code, enabling the replication of the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have included complete and executable code within the supplemental material, ensuring the reproducibility of our results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided detailed descriptions of our experimental setup in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The experimental results are presented as the mean and standard deviation over 10 runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the computational resources used for all experiments in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research adheres to the NeurIPS Code of Ethics, and we have ensured that all aspects of our work.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts in conclusion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not involve releasing data or models that pose a high risk for misuse, so no specific safeguards are necessary.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have accurately credited the source.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The supplemental material includes the file of our model's core code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects, so this information is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research does not involve human subjects, so IRB approvals or equivalent reviews are not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.