Enhancing 3D Face Analysis Using Graph Convolutional Networks with Kernel-Attentive Filters

Francesco Agnelli
University of Milan
Milano, IT
francesco.agnelli@unimi.it

Omar Ghezzi
University of Milan
Milano, IT
omar.ghezzi@studenti.unimi.it

Giorgio Blandano
University of Milan
Milano, IT
giorgio.blandano@unimi.it

Jacopo Burger
University of Milan
Milano, IT
jacopo.burger@unimi.it

Giuseppe Facchi
University of Milan
Milano, IT
giuseppe.facchi@unimi.it

Lia Schmid
University of Heidelberg
Heidelberg, DE
lia.schmid98@gmx.de

Abstract

Graph Structure Learning (GSL) techniques improve real-world graph representations, enabling Graph Neural Networks (GNNs) to be applied to unstructured domains like 3D face analysis. GSL dynamically learns connection weights within message-passing layers of a GNN, particularly in Graph Convolutional Networks (GCNs). This becomes crucial when working with small sample datasets, where methods like Transformers, which require large data for training, are less effective. In this paper, we introduce a kernel-attentive Graph Convolutional Network (KA-GCN) designed to integrate a positional bias through attention mechanisms into small-scale 3D face analysis tasks. This approach combines kernel- and attention-based mechanisms to adjust different distance measures and learn the adjacency matrix. Extensive experiments on the FaceScape public dataset and a private dataset featuring facial dysmorphisms show that our method outperforms state-of-the-art models in both effectiveness and robustness. The code is available at https://github.com/phuselab/KA-CONV.

CCS Concepts

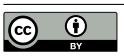
• **Computing methodologies** → **Computer vision**.

Keywords

Graph Structure Learning, 3D Face Analysis, FaceScape Dataset, Gender Recognition, Expression Recognition, Action Unit Recognition, Attention Network, Small Dataset

ACM Reference Format:

Francesco Agnelli, Omar Ghezzi, Giorgio Blandano, Jacopo Burger, Giuseppe Facchi, and Lia Schmid. 2025. Enhancing 3D Face Analysis Using Graph Convolutional Networks with Kernel-Attentive Filters. In *The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25), March 31-April 4, 2025, Catania, Italy*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3672608.3707910



SAC '25, March 31-April 4, 2025, Catania, Italy

ACM ISBN 979-8-4007-0629-5/25/03
https://doi.org/10.1145/3672608.3707910

1 Introduction

In recent decades, Graph Neural Networks (GNNs) [27] have emerged as a critical tool for learning from graph-structured data. These networks have been successfully applied across a range of domains, including computer vision [25, 28], natural language processing [26], chemistry [30], physics [21], biology [12], traffic networks [7], and recommendation systems [5].

The effectiveness of GNNs can largely be attributed to their ability to leverage the rich information embedded within both graph structure and node attributes, utilizing a message-passing mechanism to compute global information, termed states, for each local node [18]. However, real-world graphs often exhibit noise and incompleteness, posing significant challenges for GNNs in practical applications. GNN performance is highly sensitive to data quality, typically necessitating a denoised graph structure for effective representation learning [8, 14]. In certain fields, graph structures are inherently defined, such as in molecular data or physical systems. Conversely, in non-structural scenarios where the graph structure is not explicitly provided, careful consideration must be given to how the structure is constructed or learned [31]. Domains such as natural language processing and computer vision often require the deliberate formulation of graph structures.

Recent research has focused on the concept of Graph Structure Learning (GSL) [4, 8, 14], which aims to jointly learn an optimized graph structure alongside node representations. According to [31], GSL approaches are categorized into three main types: metric learning, probabilistic modeling, and direct optimization. This study focuses specifically on metric learning methods, where edge weights are determined by learning a metric function between node pairs. Conceptually, GSL can be viewed as a component of the message-passing layer within GNNs, wherein the adjacency matrix is dynamically learned along with other model parameters. Notable examples include GAT [23] and Transformer-based [20] graph convolutional layers, which exploit inter-node correlations to infer edge weights.

While the learning principles of adjacency matrices are applicable across various fields, this study emphasizes the unique characteristics of 3D facial data [3, 15, 17], aiming to identify the optimal graph structure for tasks involving 3D faces. Unlike molecular graphs, 3D faces lack a predefined graph structure. While the nodes can be effectively identified through facial anthropometric landmarks [2, 27], the challenge of determining potentially weighted

connections between these nodes remains unresolved. Unlike the 2D case [1], an additional challenge in 3D face analysis is data scarcity, particularly in real-world contexts such as medical applications, where the availability of data is often limited (e.g., rare diseases). This data limitation adversely affects models designed for large-scale tasks, such as Transformers [22], necessitating the incorporation of architectural biases to improve performance.

Our proposed approach is grounded in attention-based models due to their ability to focus on key nodes while filtering out irrelevant computations. Additionally, the inclusion of a strong inductive positional bias makes this approach well-suited for small sample-size problems. This method is based on the intuition that distant and proximal node connections can provide valuable information if treated differently [6, 10, 13, 19]. This differentiation is achieved by incorporating a kernel mechanism that learns the optimal neighbor configuration according to the task at hand.

In summary, this paper introduces a novel kernel-attentive graph convolutional layer, KA-Conv, which prioritizes the spatial relationships between facial landmarks over other node features. This methodology dynamically enhances node connections, whether proximal or distant. Furthermore, the approach is generalizable to other types of graphs where node features can be categorized into two distinct sets, with one set being more relevant. Our model is explicitly designed to perform effectively in data-scarce environments, combining a strong inductive positional bias with an attention mechanism.

The remainder of this paper is organized as follows. Section 2 formally introduces the problem formulation. Section 3 describes the proposed convolutional layer, the filters employed, and its integration into the overall model. Section 4 presents the experimental results and comparisons. Finally, Section 5 provides concluding remarks.

2 Notation and problem formulation

In the subsequent discussion, we denote a graph as a tuple $G = (V, E)$, where $v_i \in V$ indicates a vertex, $e_{ij} = (v_i, v_j) \in E \subseteq V \times V$ indicates an edge from node v_i to node v_j, and the cardinality of graph G is denoted by $N = |V|$. The neighborhood of a node v_i is denoted by the set $\mathcal{N}(v_i) = \{v_j \in V | (v_i, v_j) \in E\}$. We denote by $A \in \mathbb{R}^{N \times N}$ the adjacency matrix, where $A(i, j) > 0$ if and only if $e_{ij} \in E$, and each value of the adjacency matrix $a_{ij} \in \mathbb{R}^+$ represents the weight of the correspondent edge. In the context of 3D face graphs constructed from facial landmarks, the set V represents the 3D landmarks, that is, the points $p \in \mathbb{R}^3$ identified within the point cloud P that correspond to distinct facial features, such as eye corners, the nose tip, and other key positions.

The graph structure can be enhanced with attributes. We denote by $H \in \mathbb{R}^{N \times d}$ the matrix of node attributes, where $h_i \in \mathbb{R}^d$ represents the feature vector associated with node v_i. Similarly, edge attributes can be incorporated, with the matrix $K \in \mathbb{R}^{M \times c}$ representing the edge features, where $M = |E|$ and $k_{ij} \in \mathbb{R}^c$ denotes the feature vector for edge e_{ij}. The core of GSL is to simultaneously learn a new adjacency matrix $\tilde{A}$ and its corresponding graph representation $\tilde{H} \in \mathbb{R}^{N \times \tilde{d}}$ optimized for specific downstream tasks. The edge representation $\tilde{K} \in \mathbb{R}^{M \times \tilde{c}}$ can also be learned [31].

As stated in Section 1, here we focus on metric learning methods, where the adjacency matrices are dynamically learnt together with the other learnable weights through the message-passing layer within the GNN. Formally, metric learning approaches involve learning a metric function $\phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined on node embeddings h_i, h_j as

$$\tilde{A}(i, j) = \phi(h_i, h_j). \tag{1}$$

Depending on the choice of function ϕ, it is possible to further classify the metric approaches into two subgroups:

- *Kernel-based approaches*, where the function ϕ is modeled with a traditional kernel function. The pioneering work was AGCN [11], which exploited the generalized Mahalanobis distance by rewriting the metric function as

$$\phi(h_i, h_j) = \sqrt{(h_i - h_j)^T M (h_i - h_j)}, \tag{2}$$

$$\tilde{A}(i, j) = \exp\left(-\frac{\phi(h_i, h_j)}{2\sigma^2}\right), \tag{3}$$

  where the symmetric positive semi-definite matrix $M = WW^T$, with $W \in \mathbb{R}^{N \times N}$ is a trainable matrix.
- *Attentive approaches*, where an attention network is used to model a new graph structure based on node interactions. It is worth noticing that an attentive network such as GAT [23] can be equivalently interpreted as an attentive GSL approach. Formally, given a node v_i, the iterative aggregation and update steps in the message-passing mechanism at layer l are

$$h_i^l = \gamma\left(h_i^{l-1}, \bigoplus_{j : v_j \in \mathcal{N}(v_i)} \phi(h_i^{l-1}, h_j^{l-1})\right), \tag{4}$$

  where $\bigoplus$ denotes a differentiable, permutation invariant function, and ϕ and γ denote learnable functions. For graph-attentive convolution, equation (4) can be made explicit as

$$h_i^l = W_1 h_i^{l-1} + \sum_{v_j \in \mathcal{N}(v_i)} \alpha_{ij}^{l-1} W_2 h_j^{l-1}, \tag{5}$$

  where, with a little abuse of notation on the softmax function, the attentive coefficients at step $l - 1$ are defined as

$$\alpha_{ij}^{l-1} = \text{softmax}\left(\frac{(W_3 h_i)^T (W_4 h_j + W_5 k_{ij})}{\sqrt{d}}\right). \tag{6}$$

  and $\{W_i\}_{i=1}^5$ are trainable parameters. This leads to equating the adjacency matrix with the learnt attention coefficients:

$$A^l(i, j) = \alpha_{ij}^{l-1}, \quad \forall (i, j) \in \mathbb{R}^{N \times N}. \tag{7}$$

Furthermore, sparsity regularization may be applied to reduce computational costs. Standard approaches result in k-NN graphs, where each node has at most k neighbors, or ϵ-NN graphs, where edges above a threshold ϵ are discarded.

3 The proposed method

As motivated in Section 1, we propose an attentive GSL graph convolutional layer that leverages the informational value of node distance through a kernel-based approaches.

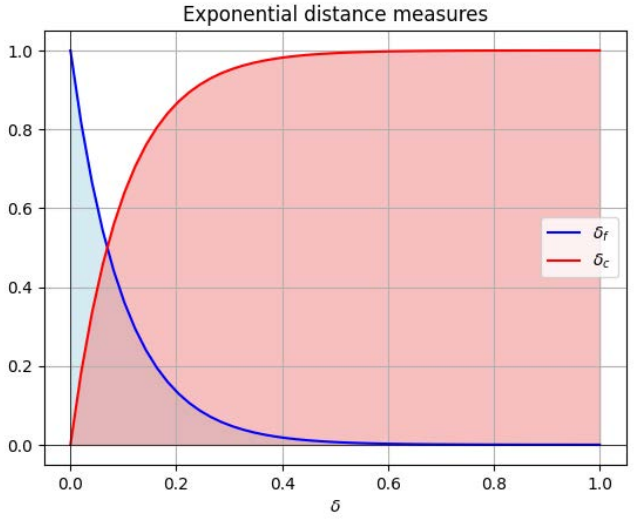


Figure 1: The kernels 'close' (equation 10) and 'far' (equation 11) as distance features.

Specifically, assume the node feature vector is given by the concatenation of two parts as

$$h_i = [\, pos_i \,||\, \tilde{h}_i \,], \tag{8}$$

where $||$ denotes the vector concatenation, $pos_i = (x_i, y_i, z_i)$ are the coordinates of the node (or, generally, a set of more relevant features), and $\tilde{h}_i = h_{i,4:d} \in \mathbb{R}^{d-3}$ are the remaining features of the node. We propose a Kernel-Attentive Graph Convolutional layer (KA-Conv), defining the metric function ϕ in equation (1) as

$$\phi(h_i, h_j) = \delta(\mathcal{A}(h_i, h_j)), \tag{9}$$

where δ is a kernel function and $\mathcal{A}$ an attentive function defined in the following section.

3.1 The Kernel-Attentive Convolutional layer

To simultaneously optimize the graph structure and its associated representations, we integrate both kernel and attention mechanisms. The kernel mechanism is designed to model the graph topology by learning a distance metric $\delta(pos_i, pos_j)$ that accurately captures the absolute positions of both nearby and distant nodes. In parallel, attention coefficients $\mathcal{A}(\tilde{h}_i, \tilde{h}_j)$ are derived from the other node features using an attentive approach. The final graph representation is achieved by effectively integrating these two approaches.

The proposed distance measure is designed to enhance the model's flexibility by allowing it to prioritize either nearby or distant nodes. This enables the model to learn and assign varying weights based on the distance between nodes. By defining the measure as the linear combination of a proximity-favoring distance and a distance-favoring measure, the model can account for all nodes while treating them differently according to their relative positions. To achieve this, we introduce two three-dimensional distance measures, a close measure and a far measure, based the exponential function.

The close and far distance metrics are defined as follows:

$$\delta^c(pos_i, pos_j) = \exp\left(-\lambda |pos_i - pos_j|\right), \tag{10}$$

and

$$\delta^f(pos_i, pos_j) = 1 - \delta^c(pos_i, pos_j), \tag{11}$$

where $\lambda \in (0, 1)$ is a hyperparameter, and $|pos_i - pos_j| = (|x_i - x_j|, |y_i - y_j|, |z_i - z_j|)$. The parameter λ intuitively controls the bias towards either nearby or distant nodes: when $\lambda \approx 0$, the majority of nodes are considered proximal; in contrast, when $\lambda >> 1$, most nodes are considered distal. Adjusting λ in this manner may enhance the network's ability to exclude certain outliers from the analysis.

The far-distance measure can be intuitively interpreted as a high-pass filter that accentuates larger distances while de-emphasizing closely connected nodes. This effectively captures significant macro-level information within the graph, such as facial dimensions and proportions. In contrast, the close-distance measure functions as a low-pass filter, focusing on the analysis of local features of the face. By employing these measures, the approach enables a clear distinction between nearby and distant nodes, analogous to the separation achieved by low-pass and high-pass filters. As shown in Figure 1, this separation is achieved with a soft, learned threshold.

The computation of each attention coefficient $z^r_{ij} = \mathcal{A}(\tilde{h}_i, \tilde{h}_j)$ involves the features $\tilde{h}_i$ and $\tilde{h}_j$, with a specific emphasis on the chosen distance measure $r \in \{c, f\}$, as

$$z^r_{ij} = \text{softmax}\left(\text{LReLU}\left(\frac{(W^r_1 \tilde{h}_i)^T W^r_2 \tilde{h}_j}{\sqrt{d^{l-1}}}\right)\right) \tag{12}$$

where $W^r_1, W^r_2 \in \mathbb{R}^{(d^{l-1}-3)\times(d^{l-1}-3)}$ are two learnable matrices that do not depend on the nodes v_i, v_j, and LReLU is an abbreviation for the LeakyReLU function.

The edge weights $\alpha^r_{ij} \in \mathbb{R}$ are determined by the product of the attention coefficient z^r_{ij} and the L2-norm of the r-distance vector as

$$\alpha^r_{ij} = z^r_{ij} ||\delta^r(pos_i, pos_j)||^2. \tag{13}$$

Optionally, to reduce computational costs, a sparsity constraint can be introduced. This can be done by defining

$$M^r = \max(\alpha^r_{ij}), \quad \forall (i, j) \in N \times N, \tag{14}$$

and using it as a threshold along with an arbitrary constant $\epsilon \in [0, 1]$, as follows:

$$\alpha^r_{ij} = \begin{cases} 0 & \text{if } \alpha^r_{ij} < \epsilon M^r \\ \alpha^r_{ij} & \text{if } \alpha^r_{ij} \geq \epsilon M^r. \end{cases} \tag{15}$$

Finally, the edge weights $\alpha^f_{ij}, \alpha^c_{ij}$ are linearly combined to obtain the final attention coefficients

$$\alpha_{ij} = l^c_{ij} \alpha^c_{ij} + l^f_{ij} \alpha^f_{ij}, \tag{16}$$

where $l^c_{ij}, l^f_{ij} \in [0, 1]$ are learnable parameters. The use of two separate parameters, instead of a convex combination, gives the model greater independence in emphasizing the most relevant component without, however, neglecting the other one altogether.

Once the kernel-attentive coefficients are settled, the updated state is derived using an attentive graph convolution as in equation

(5). Following [9], the adjacency matrix obtained is normalized to $\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, where $D_{i,i} = \sum_{j=1}^{n} A_{ij}$.

If the graph is equipped with edge features k_{ij}, we supply an integration of KA-Conv to take deeply advantage of them. In this vein, we followed the design of the Transformer convolution [20], exploiting the edge features in both the calculation of the attention and the aggregate step of message-passing.

Firstly, equation (12) is completed as

$$z_{ij}^{r} = \text{softmax}\left(\text{LReLU}\left(\frac{\left(W_1^r \tilde{h}_i\right)^T W_2^r \tilde{h}_j + W_3^r k_{ij}}{\sqrt{d^{l-1}}}\right)\right)$$

where $W_3^r \in \mathbb{R}^{1\times c}$ is a linear, learnable transformation.

For the message-passing, we integrate edge features keeping the relation between edge e_{ij} and neighbor v_j of node v_i. Unlike the Transformer approach, where a linear transformation of the edge feature is summed with the node feature, we opt to concatenate node and edge features and then apply a linear transformation. This approach should be more robust when the node and edge feature dimensions differ significantly, avoiding to overexpose the low-dimensional vector. Hence, equation (5) is rewritten as

$$h_i^l = W_4 h_i^{l-1} + \sum_{v_j \in \mathcal{N}(v_i)} \alpha_{ij}^{l-1} W_5 \left[h_j^{l-1} \,||\, k_{ij}^{l-1} \right], \qquad (17)$$

where $||$ denotes the vector concatenation on the last dimension and $W_2 \in \mathbb{R}^{(d^{l-1}+c)\times d^l}$.

Algorithm 1 describes all the steps necessary to update the state of a node v_i using the KA-Conv. In our case, the set of edges E is equal to $N \times N$. However, the whole model can be easily adapted to different choices of the edge matrix E. Since KA-Conv can be seen as a more complex version of a Transformer graph layer with the integration of biases, the computational cost is slightly higher but still quadratic in the dimension of the input (in both cases $O(d^2|E|)$. However, as pointed out in Section 1, the rationale of this model is the limited availability of 3D face data. Consequently, the increased resource usage is negligible.

3.2 The KA-GCN model

The KA convolutional layer described in Subsection 3.1 is integrated into the Kernel-Attention Graph Convolutional Network (KA-GCN) as shown in Figure 2. In the design of the architecture, the model comprises one KA-Conv layer. After that, a combination of BatchNorm and ReLU generates the node embedding for the final classification. To address overfitting, a dropout layer is added in the training part. The final classification layer is composed of an add pooling followed by a fully connected classifier.

4 Experimental results

In the experimental session, we test KA-GCN on two datasets: FaceScape [29] and FDD (a private dataset). We verify the model on different tasks (gender, emotion, and action unit recognition) and compare it with models that employ cutting-edge attentive convolutional layers (such as Transformer and GAT) as well as the baseline Graph Convolutional layer. Extastive parameter tuning and ablation studies have been conducted to identify the best model architecture for each scenario.

Algorithm 1 KA-Conv layer

Input:
- features $h_i = (pos_i, \tilde{h}_i)$, $h_j = (pos_j, \tilde{h}_j)$, k_{ij}
- hyperparameters λ, ϵ
- learnable params W_1^r, W_2^r, W_3^3, W_4, W_5, l_{ij}^r

Output: updated features h_i

for $j : e_{ij} \in E$ **do**
  $\alpha_{i,j}^c = e^{-\lambda |pos_i - pos_j|}$
  $\alpha_{i,j}^f = 1 - e^{-\lambda |pos_i - pos_j|}$
  for $r \in \{c, f\}$ **do**
    $z_{ij}^r = \text{softmax}\left(\text{LReLU}\left(\frac{\left(W_1^r \tilde{h}_i\right)^T W_2^r \tilde{h}_j + W_3^r k_{ij}}{\sqrt{d}}\right)\right)$
    $\alpha_{ij}^r = z_{ij}^r ||\alpha_{ij}^r||^2$
    $\alpha_{ij}^r = \text{Normalize}_j(\alpha_{ij}^r)$ ▷ Normaliz. in $[0, 1]$
  end for
  $\alpha_{ij} = l_{ij}^c \, \alpha_{ij}^c + l_{ij}^f \, \alpha_{ij}^f$ ▷ equation 16
  $\alpha_{ij} = \alpha_{ij} / \left(\sum_{k=1}^{n} \alpha_{ik} \sum_{k=1}^{n} \alpha_{kj}\right)^{1/2}$ ▷ Laplacian
  $h_i = W_3 h_i + \alpha_{ij} W_4 \left[h_j || k_{ij}\right]$
end for

For each graph node v_i a node feature is constructed collecting the 3D coordinates of the point, the Fast Point Feature Histograms (FPFH) [16] descriptor, and the one-hot encoding of the landmark. Edge characterization between nodes is based on geodesic paths, which are approximated using iterative edge flips and then interpolating with a 3-spline interpolation method, sampling 50 equidistant points along each path to ensure consistency.

4.1 Datasets and Experiment setup

The **FaceScape** [29] dataset provides textured 3D faces obtained from 938 subjects, and among them, 823 subjects have a complete acquisition including samples of four primary facial expressions (neutral, sadness, smile, and anger), and seven action units (dimpler, chin raiser, grin, brow raiser, brow lower, lip puckerer, lip funneler), together with gender information. This enables us to conduct three types of analysis: the binary identification of gender, detecting emotions across four categories, and recognizing action units across seven groups. The dataset is provided with 68 facial landmarks automatically extracted from each face. Figure 3 shows a representative subject example with the corresponding mesh and the landmarks extracted on the face.

FDD (Facial Dysmorphisms Dataset) is a private medical dataset acquired in our laboratory using the Vectra M3 3D imaging system. It includes 154 subjects, consisting of 108 men and 46 women, among whom a significant proportion exhibits pronounced facial dysmorphisms. Each subject is depicted by a textured 3D mesh. The age ranges between 4 and 71 years, with an average of 27 years, while expressions are neutral. For each face, 68 landmarks have been extracted using the 3DFA-GCN [24] method. This dataset

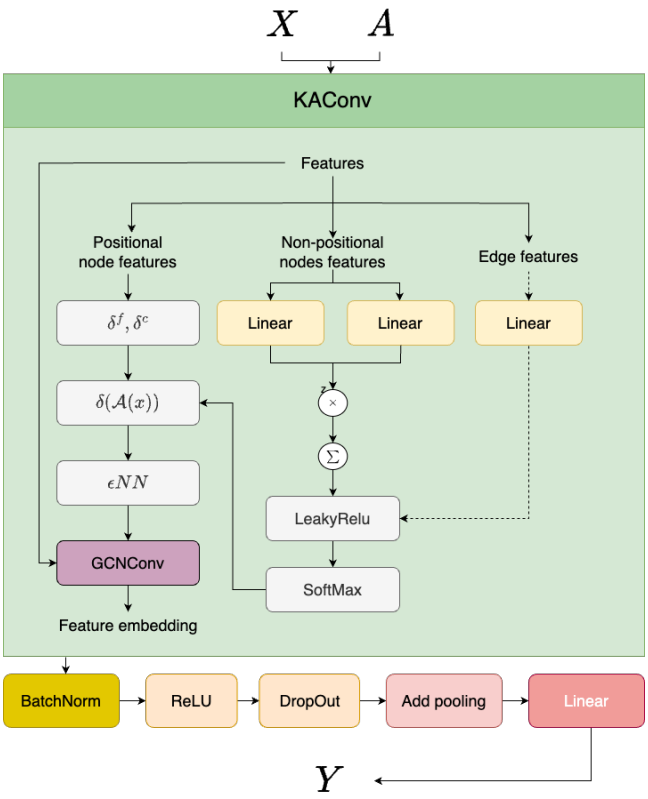


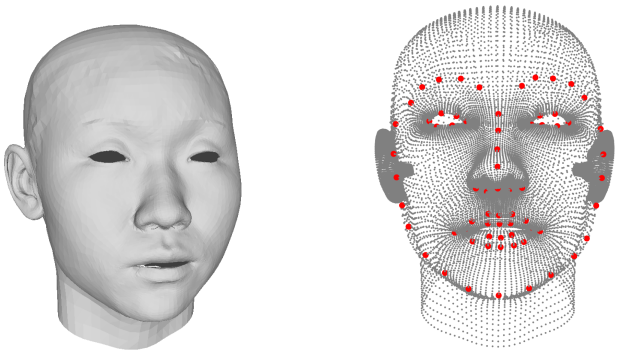
Figure 2: Architecture of the KA-GCN model. Features X and the adjacency matrix A are dynamically updated in the KA-Conv layer. Positional features are exploited by kernels 'close' and 'far' δ^f, δ^c and integrated in $\delta(\mathcal{A}(x))$ (equation (9)) with the attention coefficients calculated on the non-positional features. An ϵ-nearest neighbors operation ϵNN followed by a graph convolutional layer (GCNConv) complete the layer. The overall architecture incorporates batch normalization, a ReLU activation function, and a dropout layer. A pooling operator is then applied to produce the final embedding, which is subsequently used by a linear layer for classification.

allows us to put in place the gender classification on a challenge dataset, being unbalanced, with few samples and presenting facial dysmorphisms.

4.2 Implementation details and ablation studies

Given the distinct characteristics inherent to each task, we performed thorough parameter tuning to identify the most suitable model for each scenario. Due to the high quality of the FaceScape dataset, we have fine-tuned our models and parameters using this corpus, subsequently applying the most effective ones to the other dataset. Considering the richness of the dataset, we have chosen not to apply the ϵ-threshold to make use of all the possible information from the data.

Cross-entropy was selected as the loss function for all tasks. The models were optimized using the Adam algorithm with a learning rate of $lr = 0.001$. Both the training and testing datasets were processed in batches of 32. Each model was trained using 5-fold cross-validation over 100 epochs. For the gender classification task, the optimal configuration included a dropout rate of 0.7 and 64 hidden channels. For the emotion and action unit tasks, the highest accuracy was achieved with a dropout rate of 0.5 and 64 hidden channels.



(a) (b)

Figure 3: (a) 3D mesh sample from FaceScape; (b) Facial landmarks of a samples from FaceScape.

To evaluate the significance of both nearby and distant nodes, we conduct ablation studies comparing KA-GCN with the c-KA-GCN model, where the attention coefficients are represented as $\alpha_{ij} = \alpha^c_{ij}$ in Equation 16, and the f-KA-GCN model, where the attention coefficients are represented as $\alpha_{ij} = \alpha^f_{ij}$. Table 2 presents the results of ablation studies conducted on the FaceScape and FDD datasets for generic 3D face tasks. On the FaceScape dataset, KA-GCN consistently outperforms the other variants. Nevertheless, all models exhibit efficient performance, providing flexibility in model selection based on computational constraints. A similar trend is observed with the FDD dataset, even if here the presence of both far and close nodes become relevant to obtain high performances. This reaffirms our initial assumption that both nearby and distant nodes contribute valuable information, especially of dataset with few samples.

In a similar fashion, we conduct further ablation studies on the choice of the parameter λ from equation 10. Depending on the choice of this parameter, in fact, we obtain opposite behaviors: if $\lambda \sim 0$, the model privileges all nodes as close nodes, while choosing $\lambda >> 0$ results in a model that consider all nodes as far nodes. Table 1 presents the results of this analysis. In both datasets, the model demonstrates a tendency to favor bigger values of λ, with the optimal performance achieved at $\lambda = 10$. This value corresponds to a dense close kernel and a thinner far kernel, which tends to classify most nodes as close. All subsequent analyses are conducted using the optimal value of $\lambda = 10$.

Figure 4 illustrates the behavior of the application of the selected distance measures on the nodes weights. The transition between close and far nodes is gradual, with no clear boundary separating the two categories. This implies a smooth, continuous representation of

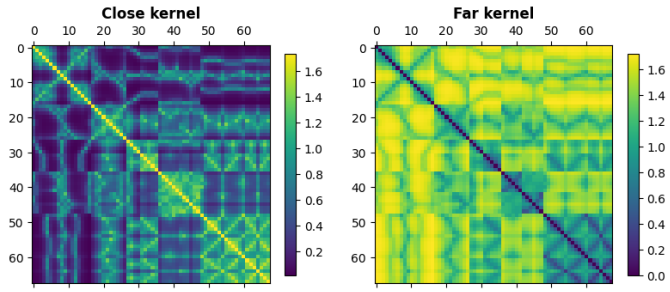


Figure 4: Distances between nodes based on the close kernel (left) and the far kernel (right) for the exponential distance measures.

proximity, where nodes can exhibit intermediate distances without a distinct threshold.

Table 1: Ablation studies varying the λ coefficient in equation (10) for KA-GCN on FaceScape (FS) and FDD dataset for different tasks.

Dataset	λ	Gender	Emotion	Action Unit
FS	1000	89.73±2.68	96.43±1.91	**99.46±0.45**
	100	90.88±2.12	97.35±1.44	99.13±0.62
	10	**92.67±1.58**	**98.89±1.12**	**99.45±0.97**
	1	91.37±2.11	97.48±0.89	99.27±0.81
	0.1	91.02±1.97	97.72±1.02	99.02±0.53
	0.01	90.74±3.62	98.24±1.03	98.63±0.43
FDD	1000	74.32±1.87	-	-
	100	74.77±1.23	-	-
	10	**76.79±2.39**	-	-
	1	75.22±2.93	-	-
	0.1	74.59±0.97	-	-
	0.01	75.05±1.32	-	-

Table 2: Ablation studies varying the distance in equation (13) for KA-Conv on FaceScape (FS) and FDD dataset for different tasks.

Dataset	Model	Gender	Emotion	Action Unit
FS	f-KA-GCN	91.57±1.69	97.98±1.56	98.81±1.04
	c-KA-GCN	91.66±1.43	98.48±2.73	98.77±0.88
	KA-GCN	**92.67±1.58**	**98.89±1.12**	**99.45±0.97**
FDD	f-KA-GCN	66.74±3.90	-	-
	c-KA-GCN	70.81±4.97	-	-
	KA-GCN	**76.79±2.39**	-	-

4.3 Comparison with SOTA models

Comparisons on multiple 3D face analysis tasks have been established to evaluate KA-Conv against state-of-the-art convolutional layers involved in graph structure learning with attention mechanisms. In particular, we have chosen a GAT convolution [23] and a Transformer convolution [20] adapted for graph structures. Furthermore, a baseline graph convolution [9] has been implemented too. To explicitly highlight the impact of each convolutional layer, all models have been constructed by sharing the architecture, as described in Section 3.2, with the convolutional layer customized accordingly. The hyperparameters for each model have been tuned on FaceScape to identify the optimal configuration for each of them in the same fashion of Section 4.2. Models trained on the FaceScape dataset were evaluated using a 5-fold Cross-Validation method across 100 epochs. In contrast, the FDD dataset, due to its smaller size, was analyzed using Leave-One-Out-Cross-Validation (LOOCV). The comparison presented in Table 3 demonstrates that KA-GCN achieves the best performance across all datasets and tasks, with the exception of the gender classification task on Facescape, where it ranks second. Clearly, the strong positional bias of KA-Conv requires high precision in landmark detection to be effective. This intuition is confirmed by the fact that for complex tasks on the FaceScape dataset, where the landmarks are provided, we outperform competitors by far more than 10%, reaching an almost perfect classification.

Table 3: Comparison between our proposal and the competitors on FaceScape (FS) and FDD dataset for different tasks.

Dataset	Model	Gender	Emotion	Action Unit
FS	GCN	82.28±4.99	76.16±5.71	71.02±6.24
	GAT	78.09±6.26	66.16±5.07	63.41±9.55
	Transformer	**91.59±5.91**	87.03±6.16	79.56±9.12
	KA-GCN	89.02±1.57	**98.87±1.39**	**98.92±0.45**
FDD	GCN	72.83±4.55	-	-
	GAT	65.12±3.70	-	-
	Transformer	71.56±5.32	-	-
	KA-GCN	**73.38±2.39**	-	-

5 Discussion and conclusions

This study examines the significance of positional biases in 3D face analysis to enhance Graph Neural Networks predictions on small datasets. We inspect geometric filters, which improve the network's capacity to capture and process the complex geometric information inherent in 3D facial structures. The combination of these filters with attentive methods leads to the definition of KA-Conv, a novel convolutional layer that incorporates node distances into metric spaces. Aligned with the widely researched Graph Structure Learning (GSL) paradigm, our model is adept at identifying graph substructures within 3D facial data.

We conducted experiments on limited datasets to determine whether the inductive positional bias contributes to robust performance despite the constraints imposed by limited data availability. Ablation studies underscore the critical importance of integrating separately proximal and distal nodes in the analysis, particularly for datasets with restricted data.

Performance variations across different datasets highlight that the accuracy of landmark locations and the overall quality of the data play pivotal roles in model performance. Notably, KA-GCN exhibits exceptional accuracy on the FaceScape dataset, outperforming competitors by over 10% in two tasks. On the FDD dataset, our method also achieves state-of-the-art results, although the performance gap is less pronounced compared to that of FaceScape.

While this study primarily focuses on 3D face analysis, the foundational principles of KA-GCN are broadly applicable to other domains involving graph structures that require dynamic learning

or optimization, especially in data-scarce environments. This approach is particularly effective when precise key points are available, allowing the model to prioritize relevant features.

References

[1] Francesco Agnelli, Giuliano Grossi, Alessandro D'Amelio, Marco De Paoli, and Raffaella Lanzarotti. 2025. VATE: a Large Scale Multimodal Spontaneous Dataset for Affective Evaluation. In *Computer Vision – ECCV 2024 Workshops*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Springer Nature Switzerland, Cham.

[2] Jacopo Burger, Giorgio Blandano, Giuseppe Maurizio Facchi, and Raffaella Lanzarotti. 2025. 2S-SGCN: A two-stage stratified graph convolutional network model for facial landmark detection on 3D data. *Computer Vision and Image Understanding* 250 (2025), 104227. https://doi.org/10.1016/j.cviu.2024.104227

[3] Jacopo Burger, Giuseppe Facchi, Giuliano Grossi, Raffaella Lanzarotti, Federico Pedersini, and Gianluca Tartaglia. 2023. Extrinsic Calibration of Multiple Depth Cameras for 3D Face Reconstruction. In *International Conference on Image Analysis and Processing*. Springer, 357–368.

[4] Bahare Fatemi, Layla El Asri, and Seyed Mehran Kazemi. 2021. SLAPS: Self-supervision improves structure learning for graph neural networks. *Advances in Neural Information Processing Systems* 34 (2021), 22667–22681.

[5] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2023. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–51.

[6] Shalini Gupta, Mia K Markey, and Alan C Bovik. 2010. Anthropometric 3D face recognition. *International journal of computer vision* 90 (2010), 331–349.

[7] Weiwei Jiang and Jiayun Luo. 2022. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications* 207 (2022), 117921.

[8] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. 2020. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 66–74.

[9] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[10] GA Kukharev and N Kaziyeva. 2020. Digital facial anthropometry: application and implementation. *Pattern Recognition and Image Analysis* 30 (2020), 496–511.

[11] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. 2018. Adaptive graph convolutional neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[12] Rui Li, Xin Yuan, Mohsen Radfar, Peter Marendy, Wei Ni, Terence J O'Brien, and Pablo M Casillas-Espinosa. 2021. Graph signal processing, graph neural network and graph learning on biological data: a systematic review. *IEEE Reviews in Biomedical Engineering* (2021).

[13] Claudio Loconsole, Catarina Runa Miranda, Gustavo Augusto, Antonio Frisoli, and Verónica Orvalho. 2014. Real-time emotion recognition novel method for geometrical facial features extraction. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, Vol. 1. IEEE, 378–385.

[14] Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen, and Xiang Zhang. 2021. Learning to drop: Robust graph neural network via topological denoising. In *Proceedings of the 14th ACM international conference on web search and data mining*. 779–787.

[15] Sabrina Patania, Giuseppe Boccignone, Sathya Buršić, Alessandro D'Amelio, and Raffaella Lanzarotti. 2022. Deep graph neural network for video-based facial pain expression assessment. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*. 585–591.

[16] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. 2008. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 3384–3391.

[17] Georgia Sandbach, Stefanos Zafeiriou, Maja Pantic, and Lijun Yin. 2012. Static and dynamic 3D facial expression recognition: A comprehensive survey. *Image and Vision Computing* 30, 10 (2012), 683–697.

[18] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.

[19] Chiarella Sforza, Gaia Grandi, Marcio De Menezes, Gianluca M Tartaglia, and Virgilio F Ferrario. 2011. Age-and sex-related changes in the normal human external nose. *Forensic science international* 204, 1-3 (2011), 205–e1.

[20] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. 2020. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509* (2020).

[21] Jonathan Shlomi, Peter Battaglia, and Jean-Roch Vlimant. 2020. Graph neural networks in particle physics. *Machine Learning: Science and Technology* 2, 2 (2020), 021001.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[23] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[24] Yuan Wang, Min Cao, Zhenfeng Fan, and Silong Peng. 2022. Learning to detect 3D facial landmarks via heatmap regression with graph convolutional network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 2595–2603.

[25] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)* 38, 5 (2019), 1–12.

[26] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, Bo Long, et al. 2023. Graph neural networks for natural language processing: A survey. *Foundations and Trends® in Machine Learning* 16, 2 (2023), 119–328.

[27] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[28] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. 2017. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5410–5419.

[29] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. 2020. FaceScape: a Large-scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[30] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI open* 1 (2020), 57–81.

[31] Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Qiang Liu, Shu Wu, and Liang Wang. 2021. Deep graph structure learning for robust representations: A survey. *arXiv preprint arXiv:2103.03036* 14 (2021).