EVA-GCN: Head Pose Estimation Based on Graph Convolutional Networks

Miao Xin*
Institute of Automation (CASIA)
Chinese Academy of Sciences
Beijing, China
miao.xin@ia.ac.cn

Shentong Mo†
Carnegie Mellon University
Pittsburgh, United States
shentonm@andrew.cmu.edu

Yuanze Lin†
Beihang Univerity
Beijing, China
linyuanze@buaa.edu.cn

Abstract

Head pose estimation is an important task in many real-world applications. Since the facial landmarks usually serve as the common input that is shared by multiple downstream tasks, utilizing landmarks to acquire high-precision head pose estimation is of practical value for many real-world applications. However, existing landmark-based methods have a major drawback in model expressive power, making them hard to achieve comparable performance to the landmark-free methods. In this paper, we propose a strong baseline method which views the head pose estimation as a graph regression problem. We construct a landmark-connection graph, and propose to leverage the Graph Convolutional Networks (GCN) to model the complex nonlinear mappings between the graph typologies and the head pose angles. Specifically, we design a novel GCN architecture which utilizes joint Edge-Vertex Attention (EVA) mechanism to overcome the unstable landmark detection. Moreover, we introduce the Adaptive Channel Attention (ACA) and the Densely-Connected Architecture (DCA) to boost the performance further. We evaluate the proposed method on three challenging benchmark datasets. Experiment results demonstrate that our method achieves better performance in comparison with the state-of-the-art landmark-based and landmark-free methods.

1. Introduction

Head pose estimation has become an active research area in recent years, as it is an essential module in many applications such as virtual reality [16], driving assistance [28] and human–computer interaction [32]. Existing methods can be roughly divided into two categories: 1) *landmark-based* approaches [17, 16] that estimate facial landmarks first then regress the pose angle accordingly, and 2) *landmark-free* approaches [35, 25, 23, 1] which estimate the head pose directly from images. Despite the latter is a hotspot in recent studies, landmark-based methods are wildly required by many real-life applications. This is because facial landmarks often serve as the input shared by multiple downstream tasks such as face alignment [3], head pose estimation [17], expression transfer [31] and so on. Moreover, certain advanced sensors [14] integrate the landmark detection. Hence, high-precision head pose estimation based on facial landmarks is quite attractive.

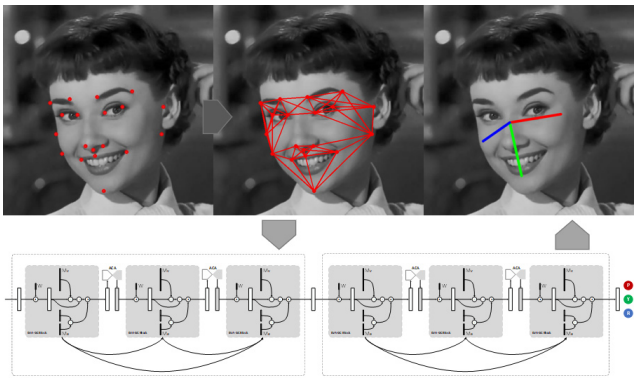

Figure 1. Given the facial landmarks, we construct a graph, and estimate the head pose utilizing the proposed EVA-GCN model. See the video demo "***Roman Holiday***" [1]. All the codes are available.

However, existing landmark-based head pose estimation methods can not present equivalent performance with the state-of-the-art landmark-free methods. The model expressive power is argued to be the main reason [21]. The principal of landmark-based methods is to achieve the 3D angle information according to the landmark distribution. Therefore, it is crucial to model the complex nonlinear relationships between the geometric distribution of landmarks and head poses robustly and efficiently. However, current methods [17, 16] are lack of the corresponding designs to fulfill such objective, resulting in the current performance bottleneck. Hence, it is natural to wonder that, ***can dedicated models designed specifically for landmark-based***

*Corresponding author.
†ShentongMo and Yuanze Lin were interns at CASIA.
[1] https://sites.google.com/view/eva-gcn

head pose estimation improve accuracy further? To answer this question, we provide a strong baseline method.

In this work, we propose to leverage the *graph convolutional networks* (GCN) [33] to improve the performance of the landmark-based head pose estimation. We propose a landmark-connection graph which takes the selected facial landmarks as the vertexes, and connect them via the k-Nearest Neighbor [38] method. We utilize the spatial GCN to regress three directions of pose angles. Specifically, we introduce the joint *edge-vertex attention*, the *automatic channel attention* and the *densely-connected architecture* in the graph convolutional networks. These designs boost the performance significantly. Our main contributions can be summarized as follows:

- We propose a graph convolutional network architecture which regresses the 3D head pose angle. To the best of our knowledge, this is the first method that introduces GCN into the head pose estimation.
- We propose joint *edge-vertex attention* mechanism into the vanilla GCN architecture, forming a strong baseline. Furthermore, we introduce the *adaptive channel attention* and the *deeply-connected architecture* into the model, improving the performance significantly.
- We evaluate the proposed method comprehensively on three challenging datasets. Our method achieves the state-of-the-art performance within the landmark-based methods and outperforms the current published landmark-free methods. We also provide the detailed ablation analysis, result discussions, and the theoretical performance bound of our method in the paper.

2. Related Work

Head Pose Estimation. Following the previous literature [35, 26], we classify the previous works into two basic categories. **Landmark-based methods** estimate the head pose using facial landmarks, geometric information or facial models. With the progress of deep learning methods, learning-based landmark detection methods [3] demonstrate superior performance. A number of landmark-based head pose estimation methods attract much attention in the research community. For example, Hyperface [24] is a multi-task model for simultaneous face detection, landmarks localization, pose estimation and gender recognition using deep convolutional neural networks (CNN). KEPLER [16] presents an iterative method for landmark prediction and pose estimation of unconstrained faces by regression. **Landmark-free methods** are proposed with the rise of the end-to-end methodology. Unlike the former ones, these methods do not have the explicit intermediate step such as landmark detection, but the mapping learning from RGB images to the pose angles is assign to the data-driven network learning. For example, FSA-Net [35] learns a fine-grained structure mapping from images for spatially grouping features before aggregation. HopeNet [26] employs very deep networks and trains using both MSE and cross-entropy losses. Thanks to the powerful non-linear feature extraction models, these methods achieve good performance in diverse situations. However, landmark-free methods have higher computational overhead and require massive data to drive the network learning. In many real-world applications, facial landmarks are important intermediate results shared by multiple downstream tasks. Therefore, it is necessary to take full advantage of landmarks to save the computation in multi-task applications.

Graph Convolutional Networks. Extending neural networks to graph data is a hotspot in the deep-learning community. GCN is applied successfully in many computer vision tasks such as skeleton-based action recognition [33], anomaly detection [39] and 3D hand shape estimation [8]. Deep GCNs are able to capture complex node interactions in graphs. However, they are hard to train owing to the vanishing gradient problem. The gated architecture [22] and the residual architecture [12] are proposed to address this problem. Recent studies introduce attention mechanism into the vanilla GCN. For instance, GAT [30] introduces vertex attention into spectral-based graph neural networks. GaAN [37] proposes a convolutional sub-network to control each attention head's importance. Edge attention is introduced into a spatial-based GCN [33]. 2s-AGCN [29] proposes an adaptive GCN to learn adaptively the graph topology.

3. Our Method

In this section, we first introduce the landmark-connection graph. Next, we introduce the graph convolutional networks with joint edge-vertex attention, automatic channel attention and densely-connected architecture. Finally, we report the implementation details.

3.1. Landmark-connection Graph Construction

We take FAN [3] as the landmark detector. Given K landmark locations, we select a vertex set $\mathbf{V} = \{v_i | i = 1, \ldots, N\}$ among them. The landmarks are selected according to the following principles:

- **Stableness**. For the same head pose presented by diverse persons, the selected landmarks should move less than the other landmarks.
- **Saliency**. For different head poses from the identical person, the selected landmarks should move markedly than the other landmarks.

We decide the landmarks by preliminary experiments (Sect. 4.2). Given the selected N vertexes (Figure 2(a)), we connect them with edges $\mathbf{E} = \{v_i v_j | i, j \in N\}$ via the

k-Nearest Neighbor [38] method. For each vertex in $\mathbf{V}$, we figure out its 5 nearest vertexes in the Euclidean space for all poses on average, and connect them to form an undirected *landmark-connection graph* $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ (Figure 2(c)).

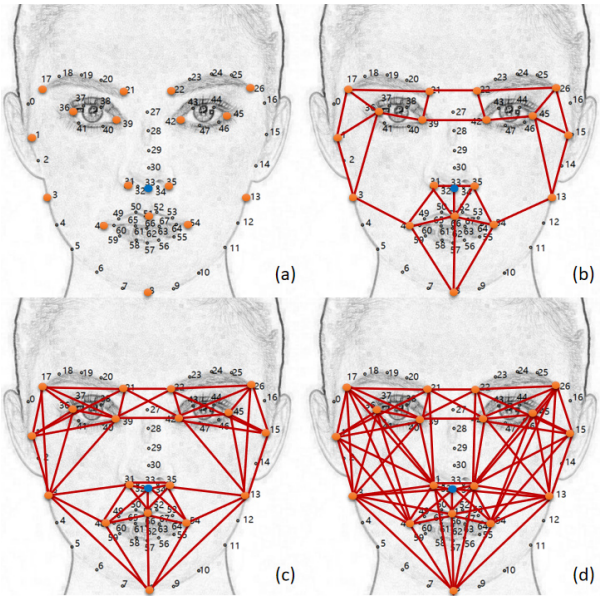

Figure 2. Landmark-connection Graphs. (a) Facial landmark locations. (b) *Graph-k3*: 3-NN connection. (c) *Graph-k5*: 5-NN connection (our final choice). (d) *Graph-k7*: 7-NN connection.

3.2. GCN with Joint Edge-Vertex Attention

We aim to leverage the graph convolutional network to mine the complex non-linear mapping between landmark-connection graphs and pose angles.

Graph convolution. The standard spatial graph convolution [33] can be represented as:

$$\mathbf{X}_{out} = \mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{X}_{in}\mathbf{W}, \tag{1}$$

where $\mathbf{A}$ is the graph adjacency matrix. $\mathbf{I}$ is an identity matrix representing the self-connections. All vertexes in the graph have the self-connection to balance the graph degree [15]. $\mathbf{\Lambda}$ is a diagonal matrix in which $\Lambda^{ii} = \sum_j (A^{ij} + I^{ij})$. $\mathbf{W}$ is a convolution weight matrix. $\mathbf{X}_{in}$ and $\mathbf{X}_{out}$ are the input and output feature maps, respectively. The Eq.1 implicitly have a vertex partition strategy [33].

Joint edge-vertex attention (EVA). Certain characteristics should be considered in the head pose estimation task. First of all, different landmarks contribute unequally to the final pose estimation results. Moreover, since distinct edges are of unique importance in predicting different directions, the weight for each edge should not be the same. Hence, we introduce attention mechanism into the graph convolution, which is composed by two parts: the *vertex attention* and the *edge attention*. The *vertex attention* is implemented by:

$$\tilde{\mathbf{X}}_{in} = \mathbf{X}_{in} \otimes \mathbf{M}_v, \tag{2}$$

where $\mathbf{M}_v$ is the attention matrix and $\otimes$ denotes element-wise multiplication between a 3D tensor and a 2D matrix. $\tilde{\mathbf{X}}_{in}$ are weighted input feature maps. Furthermore, we introduce residual learning to mitigate potential adverse effects. The residual vertex-attention operation is represented as:

$$\tilde{\mathbf{X}}_{in} = \mathbf{X}_{in} \otimes (\mathbf{M}_v + 1). \tag{3}$$

The *edge attention* is implemented by:

$$\tilde{\mathbf{A}} = \mathbf{A} \otimes \mathbf{M}_e, \tag{4}$$

where $\otimes$ denotes element-wise multiplication operation between two tensors. All elements in $\mathbf{M}_v$ and $\mathbf{M}_e$ are initialized with 1, and learned via Stochastic Gradient Descent.

Thus, the graph convolution with joint *edge-vertex attention* in one vanilla EVA-GCN cell is defined as:

$$\mathbf{X}_{out} = \mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I}) \otimes \mathbf{M}_e \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{X}_{in} \otimes (\mathbf{M}_v + 1)\mathbf{W}. \tag{5}$$

Adaptive channel attention (ACA). Inspired by the Squeeze-and-Excitation Network (SENet) [11], we introduce the channel attention into the vanilla EVA-GCN to model interdependencies between channels. This is achieved by *contextual information extraction* and *channel reweighting*.

The channel context extraction is to extract the global information representation of each channel. Unlike the SENet, the contextual information in our graph scenario is extracted by the *global average graph pooling*. Formally, given the k-th feature map $\mathbf{x}_o^k$ in $\mathbf{X}_{out} = [\mathbf{x}_o^1, ..., \mathbf{x}_o^C]$, the *global average graph pooling* is implemented by

$$z^k = \frac{1}{N}\sum_{k=1}^{N} \mathbf{x}_o^k, \tag{6}$$

where N is the number of vertexes in the graph. z^k is the global context representation of the feature map.

Next, the model learns the nonlinear interactions between channels to capture channel-wise dependencies. This is implemented by

$$\mathbf{s} = \sigma(\mathbf{W}_2\delta(\mathbf{W}_1\mathbf{z})), \tag{7}$$

where $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{r}\times C}$ and $\mathbf{W}_2 \in \mathbb{R}^{C\times\frac{C}{r}}$. r is the reduction ratio ($r = 8$, see Sect. 4.5 for the discussion). δ and σ denote the ReLU and sigmoid activation function [2].

Finally, a reweighted feature channel $\tilde{\mathbf{x}}_o^k$ is obtained by element-wise rescaling with s^k:

$$\tilde{\mathbf{x}}_o^k = \mathbf{x}_o^k \otimes s^k. \tag{8}$$

Thus, the reweighted feature maps $\tilde{\mathbf{X}}_{out} = [\tilde{\mathbf{x}}_o^1, ..., \tilde{\mathbf{x}}_o^C]$ with adaptive channel attention are achieved.

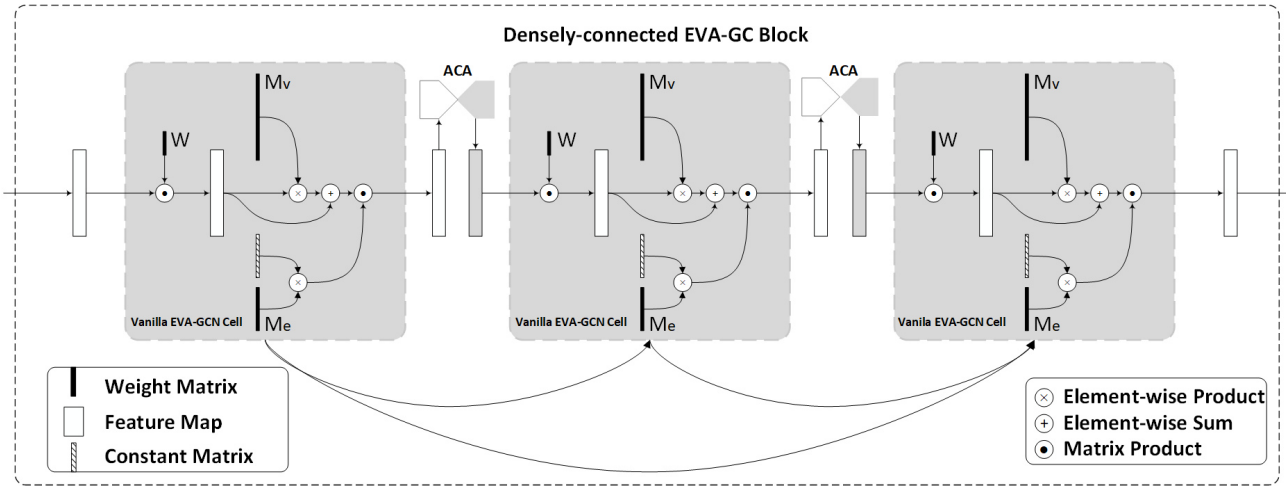

Figure 3. The architecture of the proposed densely-connected EVA-GCN with adaptive channel attention. One densely-connected EVA-GCN block contains 3 EVA-GCN cells.

Densely-connected architecture (DCA). Deeper GCNs are able to capture richer neighborhood information [18]. However, it may suffer from over-smoothing or vanishing gradient problems [10]. To improve the performance of deep EVA-GCNs, we introduce a densely-connected architecture [12] to the stacked EVA-GCN. As shown in Figure 3, the l-th block receives information from the previous blocks. Formally, the densely-connected structure can be represented as

$$\tilde{\mathbf{X}}^{l}_{out} = \mathcal{C}(\alpha^{l-1}\tilde{\mathbf{X}}^{l-1}_{out}, ..., \alpha^{1}\tilde{\mathbf{X}}^{1}_{out}), \tag{9}$$

where $\mathcal{C}$ is a concatenation operation. α is a learnable parameter which is initialized by 1. We call it the *densely-connected EVA-GCN block*, which contains 3 EVA-GCN cells with the adaptive channel attention by default.

Final model architecture. As shown in Figure 1, the network is composed of 2 densely-connected EVA-GC blocks. The first EVA-GCN cell in the first densely-connected EVA-GC block has 64 channels. The channel number is unchanged until the last block that it is enlarged to 128. The last 1×1 convolution layer is connected with a global average pooling operation to achieve a 3-dimensional output vector. We use MSE loss as the training loss function.

3.3. Implementation details

Input data and normalization. The network takes vertex features as input. The vertex feature is a 2-dimensional spatial coordinate vector $\mathbf{v}_i =< x_i, y_i >$ in the image. We normalize all vertexes to the same scale by:

$$\overline{\mathbf{v}}_i = \frac{\mathbf{v}_i - \mathbf{v}_0}{\mathbf{v}_{max} - \mathbf{v}_{min}}, \tag{10}$$

where $\mathbf{v}_0$ is the center vertex (the blue vertexes in Figure 2). $\mathbf{v}_{max}$ and $\mathbf{v}_{min}$ are the maximum and minimum values in the graph. All experiment results reported in this paper follow the same data pre-processing.

Training details. We use PyTorch [20] for implementing the proposed model. EVA-GCN is trained using Stochastic Gradient Descent (SGD) with a mini-batch size of 256. We set the initial learning rate to 0.1. The learning rate is divided by 10 at epoch $[40, 70, 90, 95]$. We terminate the training at the 100^{th} epoch. We do not use data augmentation or other tricks with bells and whistles so as to facilitate the reproduction.

4. Experiments

This section illustrates the datasets, the comparison results with state-of-the-art methods, the discussions and the ablation studies.

4.1. Dataset and evaluation protocols

We train and evaluate the proposed model on 3 challenging datasets: **300W-LP** [40], **BIWI** [7] and **AFLW2000** [40]. BIWI dataset have more than 15,000 frames. AFLW2000 dataset is the relabeled images from the AFLW dataset. 300W-LP dataset contains 61,225 images with 2D and 3D landmark annotations. These datasets have different characteristics. Most of faces in BIWI dataset are small angles (yaw: $\pm75°$, pitch: $\pm60°$, roll: $\pm50°$), while 300W-LP extend original 300-W dataset to large pose ([-90°, 90°]) but large-pose faces are generated synthetically. AFLW2000 dataset have large-pose faces and more variations in illumination and expressions. Hence, we employ these datasets to evaluate the proposed methods comprehensively. For a fair comparison with other methods, we use mean absolute error (MAE) as the evaluation metric.

4.2. Preliminary experiments

We determine the graph structure via the preliminary experiments. First, to select the salient landmarks, we calculate the moving distances of landmark locations along with the head pose changes for the identical person. As shown in Figure 4, 34 landmarks (highlight in the figure) are of larger saliency, which decides them to be the candidates.

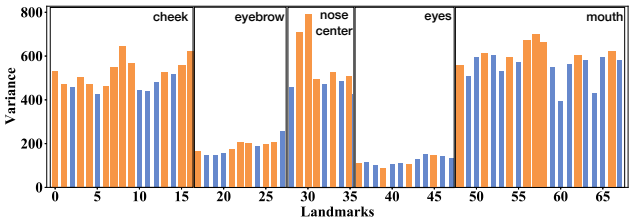

Figure 4. Comparison results *w.r.t.* the landmarks' saliency. More details and the high-resolution figure are in the supplementary.

Then, we compare the stableness of these candidates for the same head pose from diverse persons. As shown in Figure 5, we can find that the highlighted landmarks move less. Therefore, we choose these $N = 19$ landmarks (Figure 2(a)) as the vertexes of the landmark-connection graph.

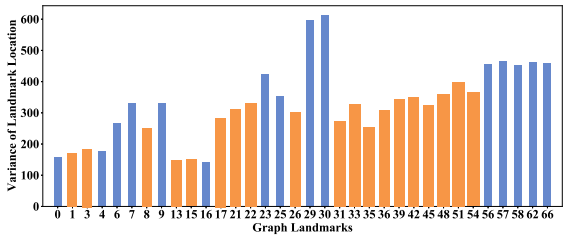

Figure 5. Comparison results *w.r.t.* the landmarks' stableness.

k **value**. Since the graph is constructed by k-Nearest Neighbor, we determine the k value by control experiments. In Figure 2, we shows other two graphs with different k value. In the control groups, *Graph-k3* has the fewest edges, so the graph structure is more sparse. *Graph-k7* is superior in information interaction between vertexes, while it's edge number is the largest. *Graph-k5* is a trade-off between *Graph-k3* and *Graph-k7*. Table 1 shows the results. Since *Graph-k5* achieves the best results in all three directions, we choose it as the final graph.

	Yaw	Pitch	Roll	MAE
Graph-k3	5.23	5.89	4.49	5.20
Graph-k7	5.18	5.72	4.35	5.08
Graph-k5	**5.13**	**5.68**	**4.31**	**5.04**

Table 1. Comparisons of different graphs for EVA-GCN performance. *Graph-k5* is selected by this experiment.

4.3. Comparison with state-of-the-art

We compare our method to the state-of-the-art methods, including landmark-based methods (**KEPLER** [16], **FAN** [3], **Dlib** [13]) and landmark-free methods (**FSA-Net** [35], **HopeNet** [26], **TriNet** [4], **SSR-Net-MD** [36], **VGG16** [9], **3DDFA** [40], **DeepHeadPose** [19]). Table 2 shows the results on AFLW2000 dataset. Our method outperforms the compared methods by a large margin.

	Yaw	Pitch	Roll	MAE
Dlib (68 points)	23.1	13.6	10.5	15.8
3DDFA	5.40	8.53	8.25	7.39
HopeNet (α=2)	6.47	6.56	5.44	6.16
HopeNet (α=1)	6.92	6.64	5.67	6.41
SSR-Net-MD	5.14	7.09	5.89	6.01
FSA-Caps (1×1)	4.82	6.19	4.76	5.25
FSA-Caps (var.)	4.96	6.34	4.78	5.36
EVA-GCN (vanilla)	5.13	5.68	4.31	**5.04**
EVA-GCN	**4.46**	**5.34**	**4.11**	**4.64**

Table 2. Comparison results with the state-of-the art methods on AFLW2000 dataset. All models are trained on 300W-LP dataset.

In these compared methods, HopeNet uses three separate losses in order to boost the performance, while we merely use one MSE loss to supervise the learning. FSA-Caps is a FSA-Net using capsule network [27] for feature aggregating. It has complex design in structure mapping learning and feature aggregation. By contrast, our model is more elegant and compact. 3DDFA is designed for large poses face alignment (up to 90°). However, its performance is not balance in a wide angle range. SSR-Net takes a coarse-to-fine strategy and performs multi-stage classification. Dlib is a tradition landmark-based method. But its weak ability in feature extraction is the main reason causing its low performance.

Generalization ability. To evaluate the generalization ability of our model, we test the model on BIWI-test dataset directly without fine-tuning. The results are shown in Table 3. The compared methods include not only landmark-based methods, but also landmark-free methods. Our method outperforms all state-of-the-art landmark-based methods (FAN, KEPLER) and the current best landmark-free model (FSA-Caps). A very recent work, img2pose [1], utilizes a super large face recognition dataset [34] to pre-train the model and achieves MAE=3.90 (Yaw:3.97, Pitch:5.27, Roll:2.46) one the head pose estimation task. We do not use auxiliary datasets but the result is comparable.

Furthermore, we retrain the model on BIWI-train dataset and evaluate it on BIWI-test dataset to know how good performance it can achieve. We use 70% of videos in the BIWI dataset for training and the rest videos for testing. Table 4 shows the results. MAE of our method is lower than FSA-Net by 0.56, which means that the advantage is more

significant if we fine-turn the model using given datasets. The results show the method's good generalization capability across datasets. Moreover, since BIWI dataset is composed by synthetic data, this result suggests that our method can utilize the existing data better and is robust to the *domain shift*, meaning that the cost in data acquisition is lower.

	Yaw	Pitch	Roll	MAE
Dlib	16.8	13.8	6.19	12.2
KEPLER	8.80	17.3	16.2	13.9
FAN	8.53	7.48	7.63	7.89
3DDFA	36.2	12.3	8.78	19.1
FSA-Caps (1×1)	4.78	6.24	3.31	4.31
FSA-Caps (var.)	4.56	5.21	3.07	4.28
EVA-GCN (vanilla)	3.90	5.31	3.02	**4.08**
EVA-GCN	**4.01**	**4.78**	**2.98**	**3.92**

Table 3. Comparison results with the state-of-the art methods on BIWI-test dataset. All models are trained on 300W-LP dataset.

	Yaw	Pitch	Roll	MAE
DeepHeadPose	5.67	5.18	-	-
SSR-Net-MD	4.24	4.35	4.19	4.26
VGG16	3.91	4.03	3.03	3.66
FSA-Caps-Fusion	2.89	4.29	3.60	3.60
TriNet	2.93	3.04	2.44	2.80
EVA-GCN (vanilla)	3.39	3.97	2.59	**3.32**
EVA-GCN	**2.01**	**2.82**	**1.89**	**2.24**

Table 4. Comparison results with the state-of-the art methods on BIWI-test dataset. All models are trained on BIWI-train dataset.

4.4. Discussion

Stability. To evaluate the prediction stability of our method, we enumerates the per-frame pose estimation errors on a 492-frame video in Figure 6. Comparing to other methods, our method is more stable, and the predicted results are more smooth. This nature is attractive for certain applications in virtual reality (VR).

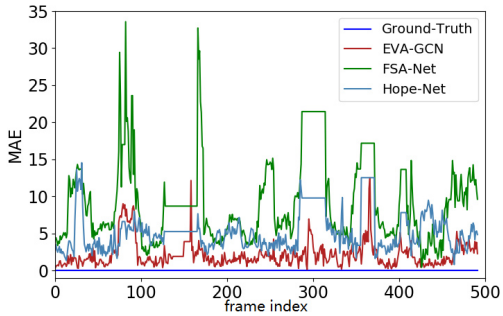

Figure 6. Visualization on a 492-frame video from BIWI dataset (24^{th} video). Our method (red curve) is more stable.

Case analysis. To evaluate our method comprehensively, we analyze the failure and successful cases by detailed statistics. Figure 7 shows a few of cases. We calculate the failure cases on BIWI-test dataset (error threshold is $\pm5°$) in $10°$ per group. The results on three directions are plotted as three curves in Figure 8. Our failure cases are relatively less and distributed mostly in large poses ($> |65°|$). However, checking the cause of errors, we find that the main reason is that the face detectors perform unsatisfactorily on some large-pose examples. Less than 2% of large-pose faces can not be detected from the images, so the head pose estimators receive some incorrect face regions. Actually, all compared methods suffer from this issue (e.g., FSA-Net's results in Figure 7). Hence, we argue that more robust face detectors can help head pose estimators achieve better performance.

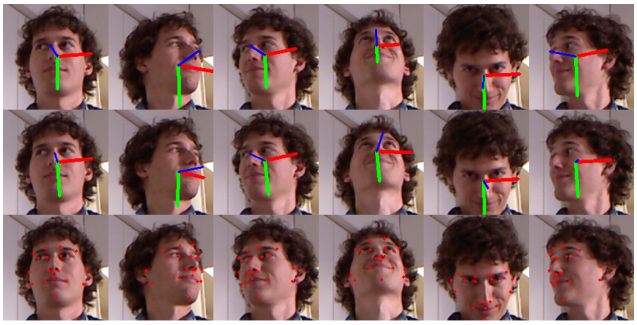

Figure 7. *Top*: EVA-GCN results; *Middle*: FSA-Net results; *Bottom*: Detected landmarks.

Speed and model size. The interference speed of EVA-GCN is 586 FPS on one Nvidia Titan RTX GPU. This suggests that the time consuming caused by the EVA-GCN is almost negligible, and the final speed of the head pose estimation is up to the landmark detector. The EVA-GCN with the FAN landmark detector achieves 56 FPS. Our method achieves nearly the same speed as the FSANet (58 FPS). Our model is only 1.03 MB of size. Comparing to FSA-Net and HopeNet (2.9 and 95.9 MB), it is light-weight. Note that our model do not have any compression, pruning or other engineering optimizations, so there is still much room for efficiency boosting. This result suggests that the proposed model enjoys high efficiency and good performance.

We also implement an end-to-end EVA-GCN with CNN-based landmark detection and our GCN-based pose estimation. It achieves 82 FPS and MAE = 4.92 on AFLW2000. We also implement it on a low-power ARM edge computing device, which achieves 45 FPS. Note that this is beyond the scope of this paper. More details can be found in the supplementary material.

4.5. Ablation study

We examine the effectiveness of the proposed components in EVA-GCN, all models are trained on 300W-LP dataset, and then tested on AFLW2000 dataset.

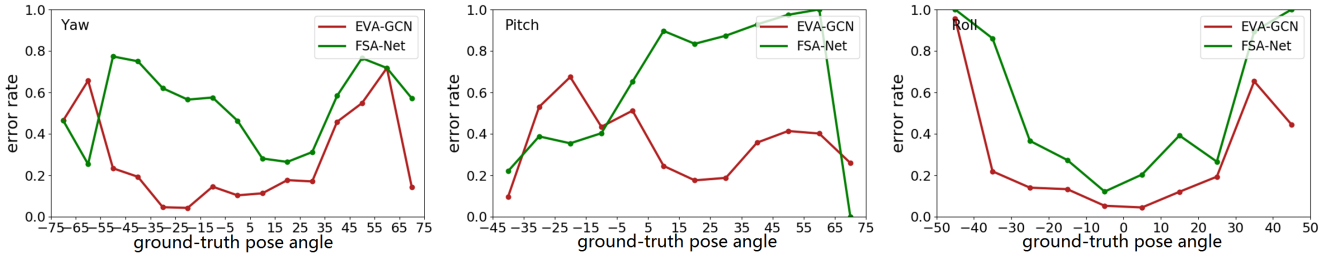

Figure 8. Error distributions of different methods on BIWI-test dataset.

Edge-vertex attention and visualization. Table 5 shows the result that compares the EVA-GCN with the standard GCN. EVA-GCN (v.) denotes the vanilla EVA-GCN (without ACA and DCA). We can find the *edge-vertex attention* improves the GCN effectively. To understand the effect of the *edge-vertex attention* mechanism intuitively, we visualize the learned attention in Figure 9. The edges and vertexes with lighter color represent their importance is larger. The distribution of the learned attention weights is uneven among edges and vertexes. We are hardly able to design it by hand without data-driven parameter learning. Although seemingly complex, we can still find some characteristics in attention distribution. For example, the *edge-attention* weights are almost distributed symmetrically. The importance of certain vertexes are relatively lower. These results are consistent with the preliminary experiments.

	Yaw	Pitch	Roll	MAE
GCN	4.93	5.89	4.75	5.19
EVA-GCN (v.)	5.13	5.68	4.31	**5.04**

Table 5. Ablation analysis: the edge-vertex attention.

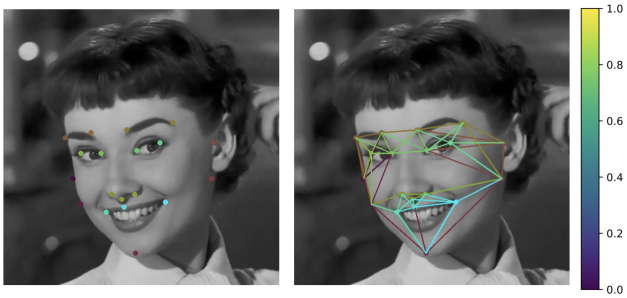

Figure 9. Vertex and edge attention visualization.

Adaptive channel attention and the reduction ratio. To evaluate the effectiveness of the proposed *adaptive channel attention*, we ablate it from the proposed model to observe the results. As shown in Table 6, we can find that the adaptive channel attention mechanism improve the performance significantly. Since r is a hyper-parameter, we conduct experiments *w.r.t.* a range of r values. Seen from Table 6, the model achieves the best result when $r = 8$. Therefore, r is determined to be 8.

Densely-connected architecture. We compare the vanilla EVA-GCN with the vanilla EVA-GCN + densely-connected architecture. As reported in Table 7, we can find that the densely-connected architecture can boost the performance effectively.

	Yaw	Pitch	Roll	MAE
EVA-GCN(v.)	5.13	5.68	4.31	5.04
EVA-GCN(v.)+ACA(r=2)	5.03	5.48	4.23	4.91
EVA-GCN(v.)+ACA(r=4)	4.96	5.42	4.19	4.86
EVA-GCN(v.)+**ACA**(r=8)	**4.86**	**5.39**	**4.17**	**4.81**
EVA-GCN(v.)+ACA(r=16)	4.92	5.46	4.22	4.87

Table 6. Ablation analysis: MAE across different reduction ratios.

	Yaw	Pitch	Roll	MAE
EVA-GCN (v.)	5.13	5.68	4.31	5.04
EVA-GCN (v. + **DCA**)	4.98	5.56	4.22	4.92
EVA-GCN	4.46	5.34	4.11	4.64

Table 7. Ablation analysis: the densely-connected architecture.

Network configuration. We explore diverse network configurations with different network depth. Table 9 shows the results. In general, deeper networks have better performance, and the deeply-connected structure performs positive effect to the final results. Nevertheless, since head pose estimation is a speed-sensitive task, we argue that EVA-GCN with 2 densely-connected EVA-GC blocks (6 layers) is acceptable for most applications.

Summary. We can summarize the above ablation studies. Seen from Table 8, with all aforementioned designs, our method boost the performance consistently. Figure 10 shows the improvement ratios of each components.

4.6. Performance bound analysis

Since the landmark is an important factor in landmark-based pose estimation, we choose several mainstream detectors, including OpenPose [5], Dlib [13], RetinaFace (reproductive) [6] and FAN [3], and extract landmarks on 300W-LP dataset. We train the EVA-GCN using these landmarks results, and test the model on AFLW2000. The result is shown in Table 10. We can find that the landmark detector is important for the final results. However, even if using relatively weak detectors (e.g., Dlib), our method is still comparable to some other methods, which reflects the robustness of our method to the landmark detection errors.

GCN	vanilla EVA-GCN	vanilla EVA-GCN + ACA	EVA-GCN (ACA+DCA)	Deeper EVA-GCN	**MAE**
✓					5.19
✓	✓				5.04 (↓0.15)
✓	✓	✓			4.81 (↓0.23)
✓	✓	✓	✓		4.64 (↓0.17)
✓	✓	✓	✓	✓	4.59 (↓0.05)

Table 8. Influence of each component for the final model performance.

	Yaw	**Pitch**	**Roll**	**MAE**	**MB**
EVA-GCN (v., 6 layers)	5.11	5.64	4.29	5.01	0.98
EVA-GCN (v., 9 layers)	5.07	5.58	4.27	4.97	1.31
EVA-GCN (w/o ACA, 6 layers)	4.87	5.48	4.21	4.85	1.03
EVA-GCN (w/o ACA, 9 layers)	4.81	5.34	4.13	4.76	1.36
EVA-GCN (w/o DCA, 6 layers)	4.92	5.46	4.19	4.86	0.98
EVA-GCN (w/o DCA, 9 layers)	4.85	5.38	4.15	4.79	1.31
EVA-GCN (6 layers) (final choice)	4.46	5.34	4.11	4.64	1.03
EVA-GCN (9 layers)	4.41	5.27	4.09	**4.59**	1.36

Table 9. Performance of various EVA-GCNs with different depth. *Graph-5k* is used in this experiment.

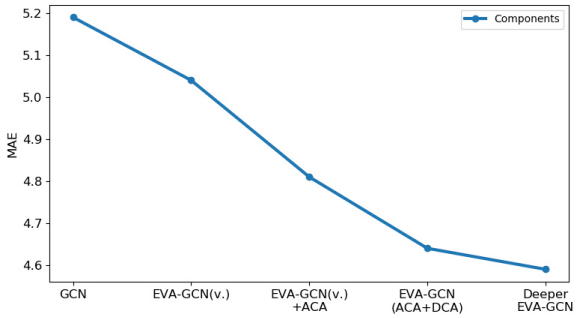

Figure 10. Improvement ratios of each components in the model for the final performance.

Landmark Detectors	**Yaw**	**Pitch**	**Roll**	**MAE**
EVA-GCN + OpenPose	7.25	5.52	4.78	5.85
EVA-GCN + Dlib	6.39	5.76	3.63	5.26
EVA-GCN + RetinaFace	5.02	5.33	4.26	4.87
EVA-GCN + FAN (ours)	4.96	5.34	4.11	4.64
EVA-GCN + **GT***	3.23	4.15	3.05	**3.48**

Table 10. Comparisons of different landmark detectors for EVA-GCN performance. GT* means ground-truth data.

To further evaluate the model's robustness to imprecise landmarks, we artificially introduce various degrees of stochastic noise to the ground-truth landmarks. For arbitrary m landmarks, we shift them randomly in a $2l \times 2l$ (pixels) square region that takes the ground-truth locations as the centers (Figure 11). Then we evaluate the results achieved by our model. We test two groups ($m = 3, 6$), and the offset range l is in $[0, 4, 8, 12, 16]$ for each group. Seen from the histograms in Figure 12, we can find that the model is robust to the noise. When 16% landmarks are shifted by less than 10 pixels, the model's performance is barely affected. Even when 32% landmarks deviated their ground truth locations by no more than 16 pixels, the EVA-GCN is still able to achieve the state-off-the-art accuracy. The GCN and attention mechanism has a positive effect in error tolerance, which is verified by other works [33, 30] as well. That forms the strong expressive power of our model.

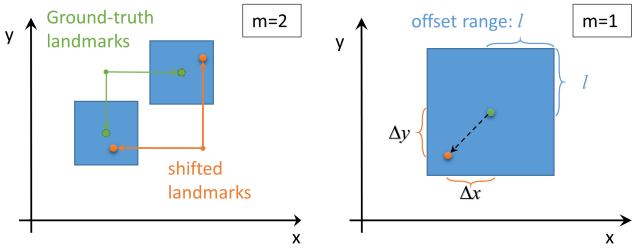

Figure 11. Landmarks with stochastic noise.

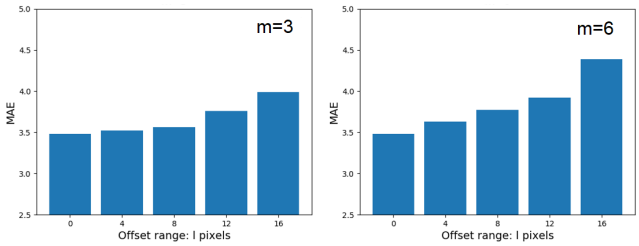

Figure 12. Performance changes with various degrees of noise. Horizontal axis: offset $l \in [0, 4, 8, 12, 16]$. Vertical axis: MAE.

Performance bound. Finally, we report the result achieved by using the ground-truth landmark labels (GT*) in Table 10. Therefore, **MAE = 3.48** (on AFLW2000 dataset) can be viewed as the **theoretical performance upper bound** of our method.

5. Conclusion and Future Works

In this paper, we propose a GCN-based head pose estimation method. We present a novel method and achieve state-of-the-art performance. Our new method gives a response to the question raised at the beginning of this paper: landmark-based methods are still worth to explore and it remains an open problem. For future works, we argue that more efficient and hybrid CNN-GCN are worth expecting. We hope this initial work can inspire more researchers.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (Nos. 61906195, 61906193).

References

[1] Vitor Albiero, Xingyu Chen, Xi Yin, Guan Pang, and Tal Hassner. img2pose: Face alignment and detection via 6dof, face pose estimation. *arXiv preprint arXiv:2012.07791*, 2020.

[2] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. A survey on modern trainable activation functions. *Neural Networks*, 2021.

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d and 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *ICCV*, 2017.

[4] Zhiwen Cao, Zongcheng Chu, Dongfang Liu, and Yingjie Chen. A vector-based representation to enhance head pose estimation. In *WACV*, 2021.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[6] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020.

[7] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *IJCV*, 101(3):437–458, 2013.

[8] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *CVPR*, 2019.

[9] Jinwei Gu, Xiaodong Yang, Shalini De Mello, and Jan Kautz. Dynamic facial analysis: From bayesian filtering to recurrent neural network. In *CVPR*, 2017.

[10] Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. Densely connected graph convolutional networks for graph-to-sequence learning. *TACL*, 7:297–312, 2019.

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[13] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014.

[14] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *CVPR Workshops*, 2017.

[15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[16] Amit Kumar, Azadeh Alavi, and Rama Chellappa. Kepler: keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors. In *IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, 2017.

[17] Stéphane Lathuilière, Rémi Juge, Pablo Mesejo, Rafael Munoz-Salinas, and Radu Horaud. Deep mixture of linear inverse regressions applied to head-pose estimation. In *CVPR*, 2017.

[18] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *ICCV*, 2019.

[19] Sankha S Mukherjee and Neil Martin Robertson. Deep head pose: Gaze-direction estimation in multimodal video. *T-MM*, 17(11):2094–2107, 2015.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, 2019.

[21] Massimiliano Patacchiola and Angelo Cangelosi. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 71:132–143, 2017.

[22] Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. Semi-supervised user geolocation via graph convolutional networks. In *ACL*, 2018.

[23] Rajeev Ranjan, Shalini De Mello, and Jan Kautz. Lightweight head pose invariant gaze tracking. In *CVPR Workshops*, 2018.

[24] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *T-PAMI*, 41(1):121–135, 2017.

[25] Nataniel Ruiz, Eunji Chong, and James M Rehg. Fine-grained head pose estimation without keypoints. In *CVPR Workshops*, 2018.

[26] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. In *CVPR Workshops*, June 2018.

[27] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *NeurIPS*, 2017.

[28] Anke Schwarz, Monica Haurilet, Manuel Martinez, and Rainer Stiefelhagen. Driveahead-a large-scale driver head pose dataset. In *CVPR Workshops*, 2017.

[29] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *ICLR*, 2017.

[31] Wei Wang, Xavier Alameda-Pineda, Dan Xu, Pascal Fua, Elisa Ricci, and Nicu Sebe. Every smile is unique: Landmark-guided diverse smile generation. In *CVPR*, 2018.

[32] Yujia Wang, Wei Liang, Jianbing Shen, Yunde Jia, and Lap-Fai Yu. A deep coarse-to-fine network for head pose estimation from synthetic data. *Pattern Recognition*, 94:196–206, 2019.

[33] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[34] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *CVPR*, 2016.

[35] Tsun-Yi Yang, Yi-Ting Chen, Yen-Yu Lin, and Yung-Yu Chuang. Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *CVPR*, 2019.

[36] Tsun-Yi Yang, Yi-Hsuan Huang, Yen-Yu Lin, Pi-Cheng Hsiu, and Yung-Yu Chuang. Ssr-net: A compact soft stage-

wise regression network for age estimation. *IJCAI*, 5(6), 2018.

[37] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit Yan Yeung. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. In *UAI*, 2018.

[38] Youqiang Zhang, Guo Cao, Bisheng Wang, and Xuesong Li. A novel ensemble method for k-nearest neighbor. *Pattern Recognition*, 85:13–25, 2019.

[39] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H. Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *CVPR*, 2019.

[40] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *CVPR*, 2016.