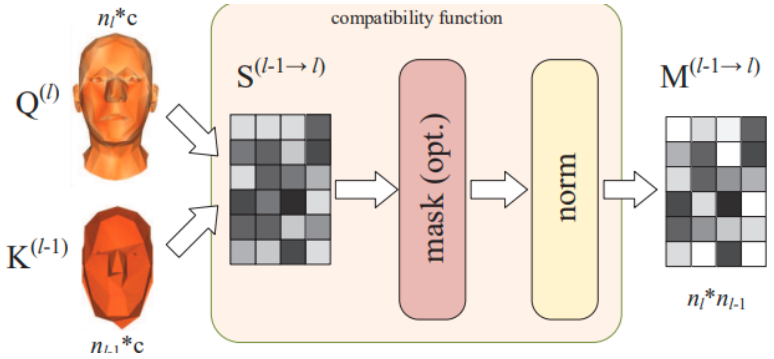
Applications of Graph Convolutional Networks (GCN)

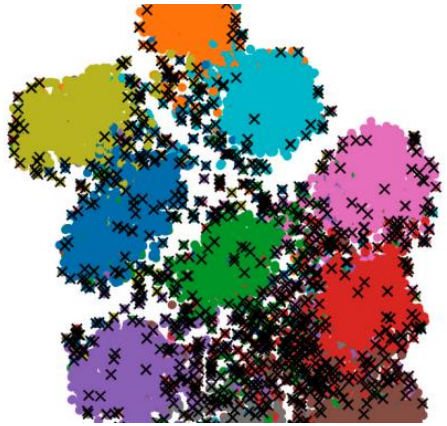
Tae-Kyun (T-K) Kim
Computer Vision and Learning Lab

KAIST Imperial College London

icvl

https://sites.google.com/view/tkkim/

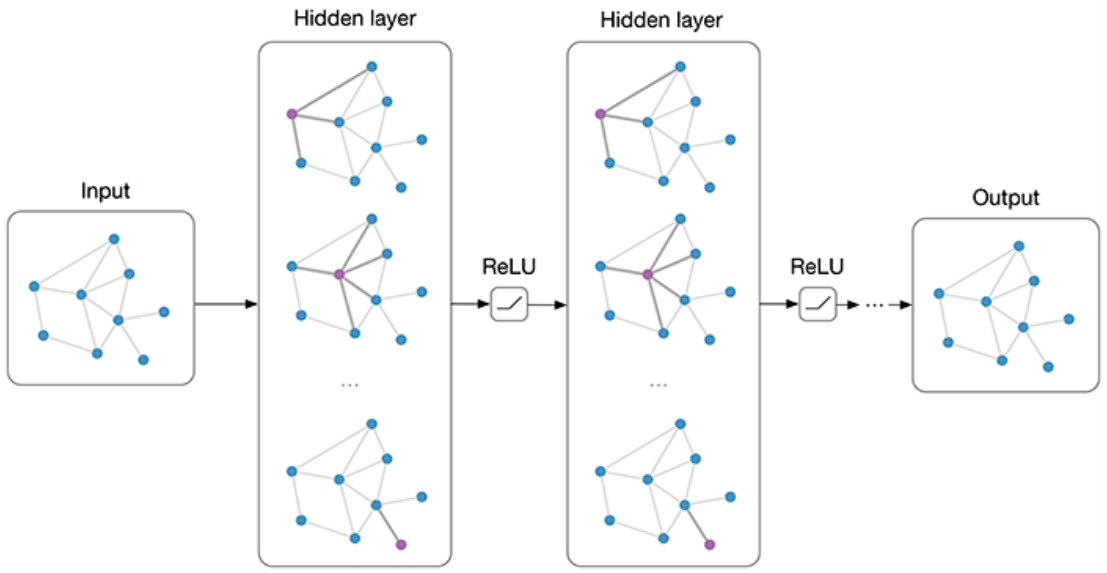




Z. Chen, T-K. Kim, Learning Feature Aggregation for Deep 3D Morphable Models, Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

R. Caramalau, B. Bhattarai, T-K. Kim, Sequential Graph Convolutional Network for Active Learning, Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

T. Kipf et al, Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017



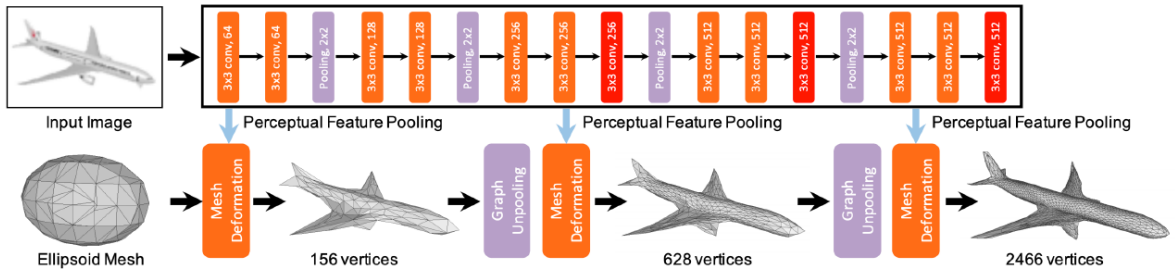
Graph convolution:

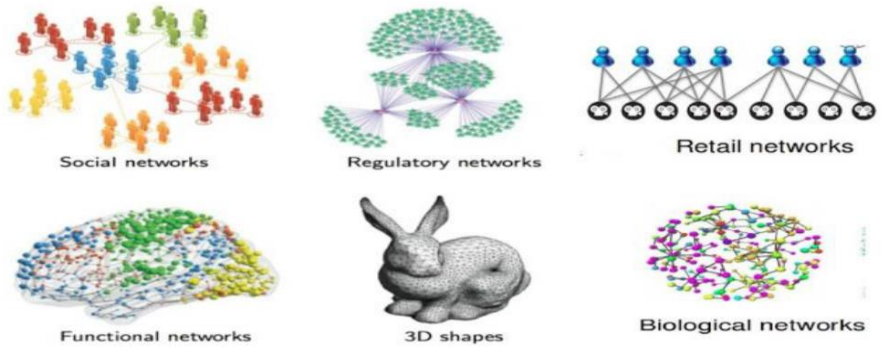
- A feature description every node in a matrix X
- An adjacency matrix A

$$H^{(l+1)} = f(H^{(l)}, A)$$
$$= \sigma\left(AH^{(l)}W^{(l)}\right)$$

Where $H^{(0)} = X$

↳ Wang et al, Pixel2Mesh, ECCV 2018





Learning Feature Aggregation for Deep 3D Morphable Models

Zhixiang Chen[1]

Tae-Kyun Kim[1,2]

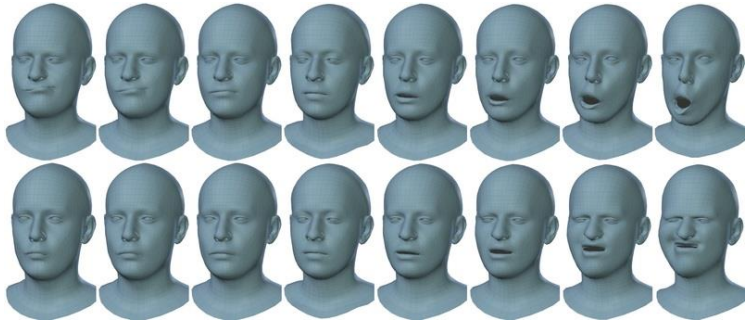
Imperial College London

KAIST

Introduction

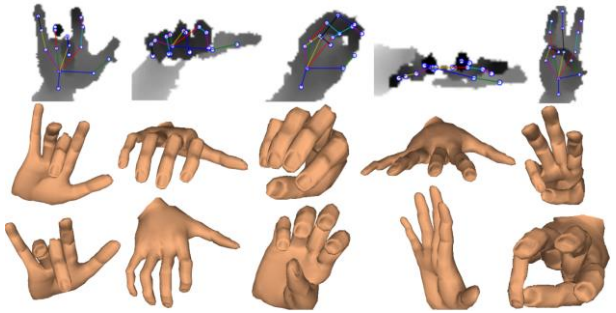
- 3D Morphable Models (3DMMs): models for registered 3D meshes of an object class, like face, body and hand
- Deep 3DMMs apply graph convolutional network (GCN) on meshes. A hierarchical mesh autoencoder is used to obtain the latent embeddings.

Goal: learning latent embeddings for registered meshes of an object class

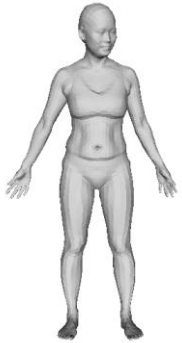


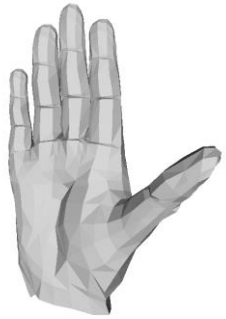
https://coma.is.tue.mpg.de/

https://dfaust.is.tue.mpg.de/



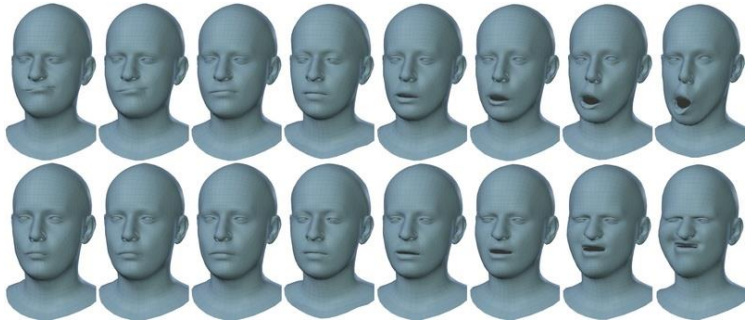
Malik *et al.* DeepHPS, 3DV 2018





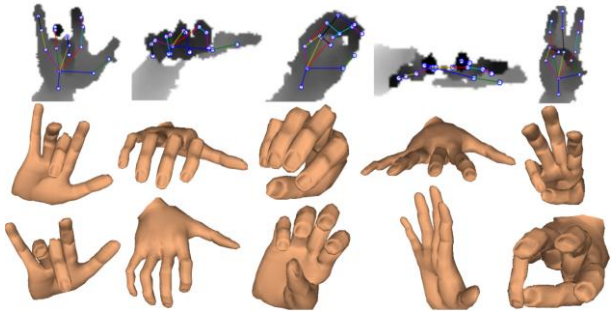
Registered meshes: 3D meshes sharing the same topology of the template mesh

Goal: learning latent embeddings for registered meshes of an object class

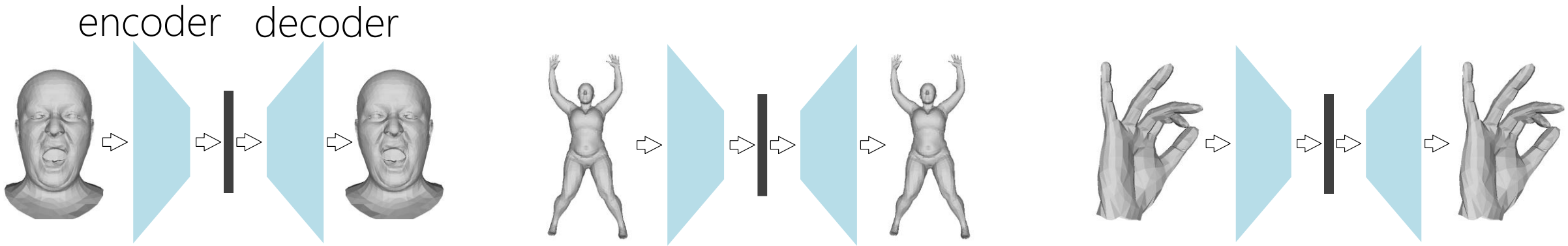


https://coma.is.tue.mpg.de/

https://dfaust.is.tue.mpg.de/



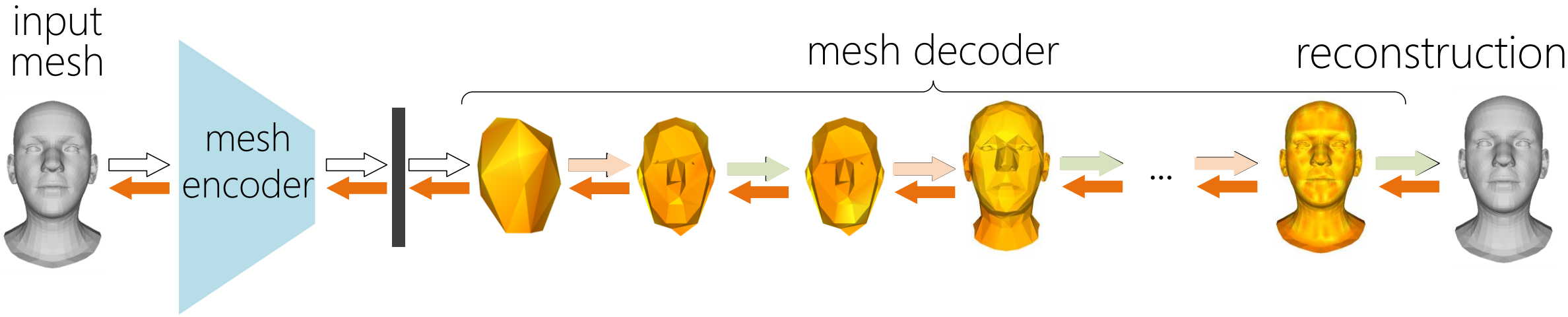
Malik *et al.* DeepHPS, 3DV 2018

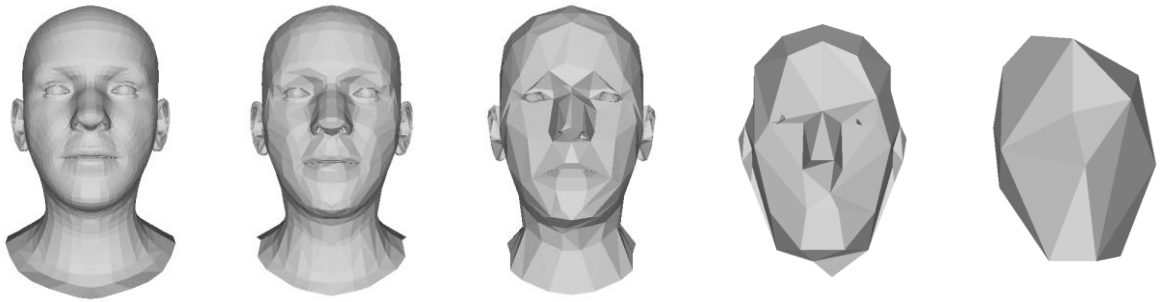


Autoencoder for learning latent embeddings of general objects

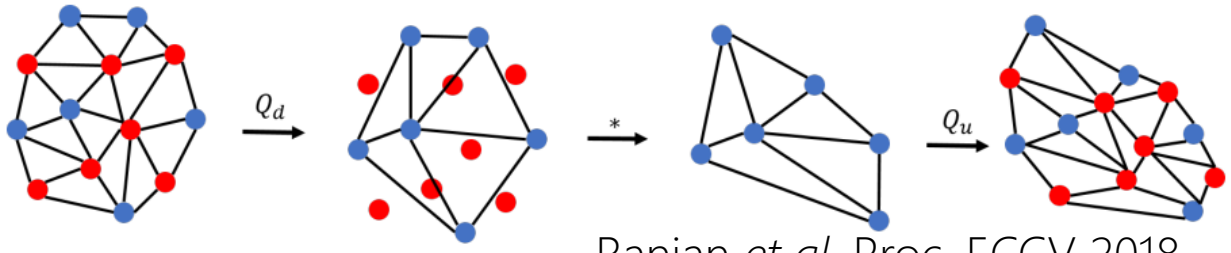
In the literature, the hierarchy is built by *mesh decimation* by minimizing surface error before and after decimation, and can be represented by mapping matrices across neighboring levels.

Autoencoding registered meshes



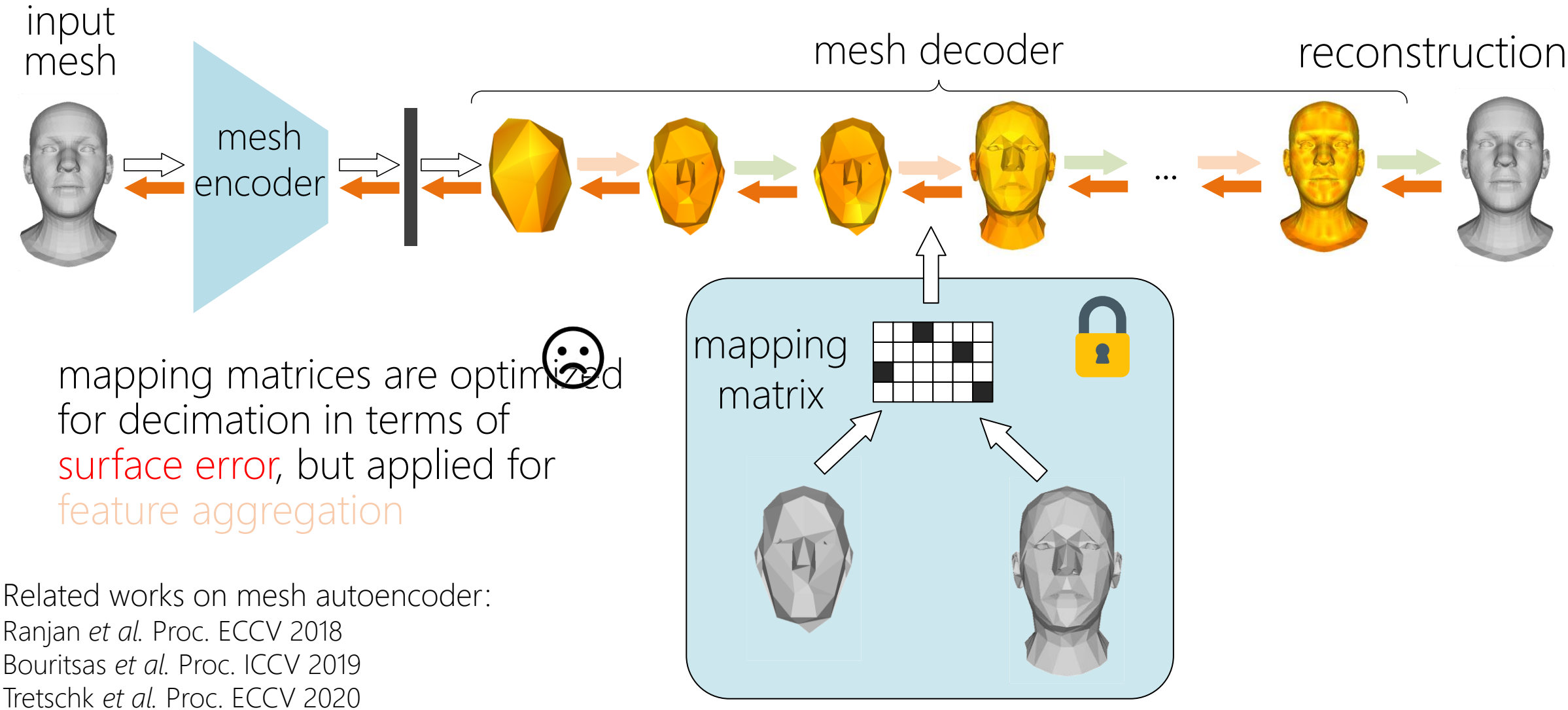


Related works on mesh autoencoder:
Ranjan *et al.* Proc. ECCV 2018
Bouritsas *et al.* Proc. ICCV 2019
Tretschk *et al.* Proc. ECCV 2020
Zhou *et al.* Proc. NeurIPS 2020



Ranjan *et al.* Proc. ECCV 2018

Autoencoding registered meshes

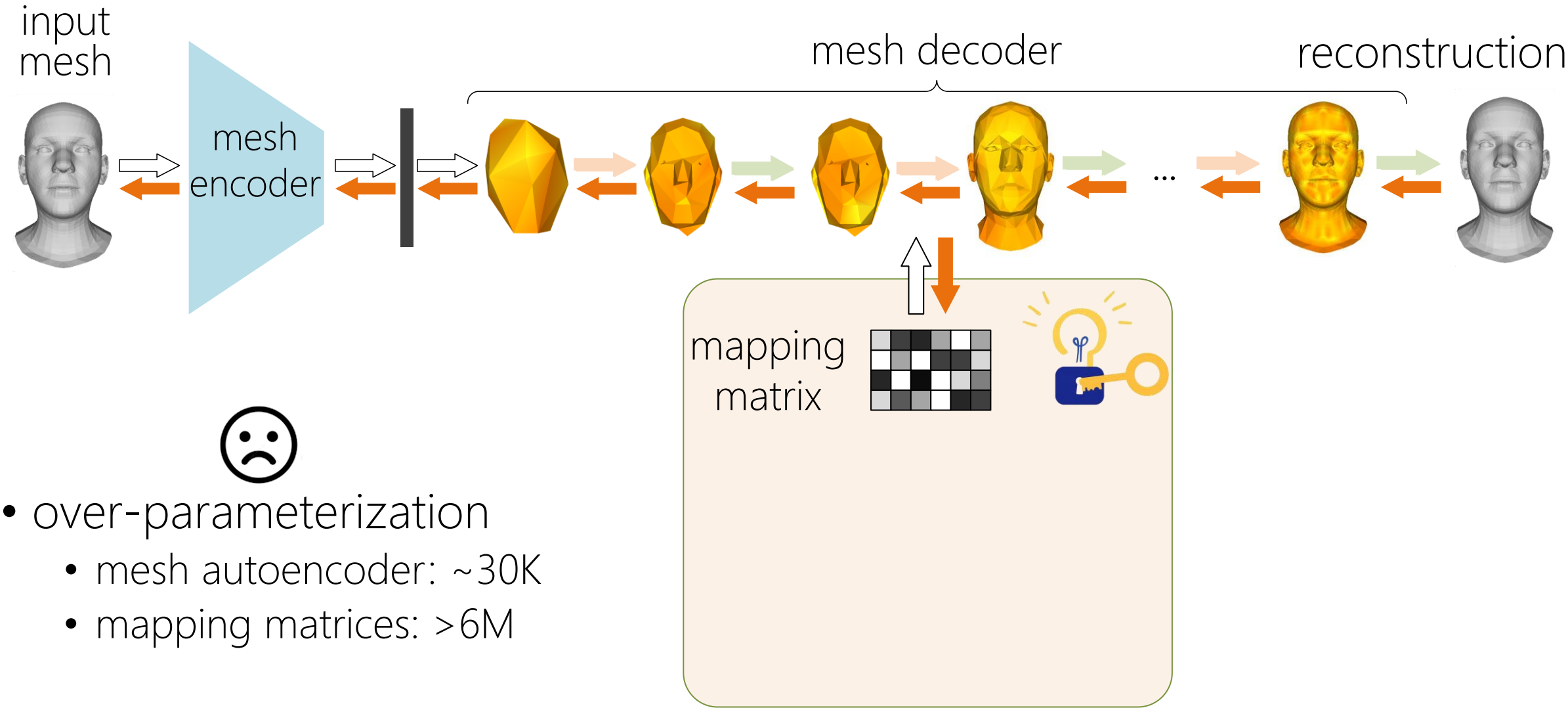


mapping matrices are optimized for decimation in terms of surface error, but applied for feature aggregation

Related works on mesh autoencoder:
Ranjan *et al.* Proc. ECCV 2018
Bouritsas *et al.* Proc. ICCV 2019
Tretschk *et al.* Proc. ECCV 2020
Zhou *et al.* Proc. NeurIPS 2020

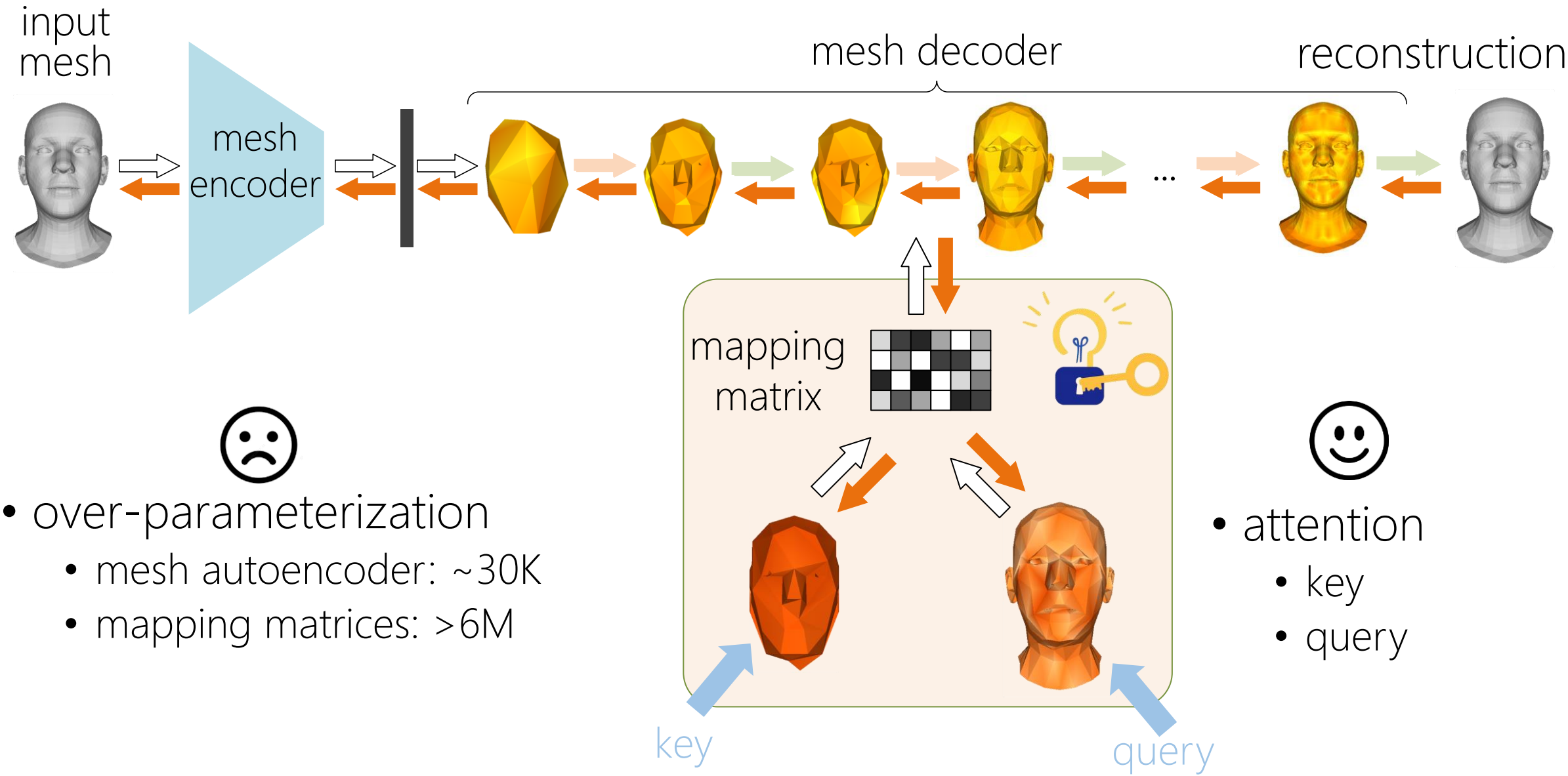
We propose to learn the mapping matrices end-to-end.

Attention based feature aggregation



- over-parameterization
  - mesh autoencoder: ~30K
  - mapping matrices: >6M

Attention based feature aggregation



- over-parameterization
  - mesh autoencoder: ~30K
  - mapping matrices: >6M

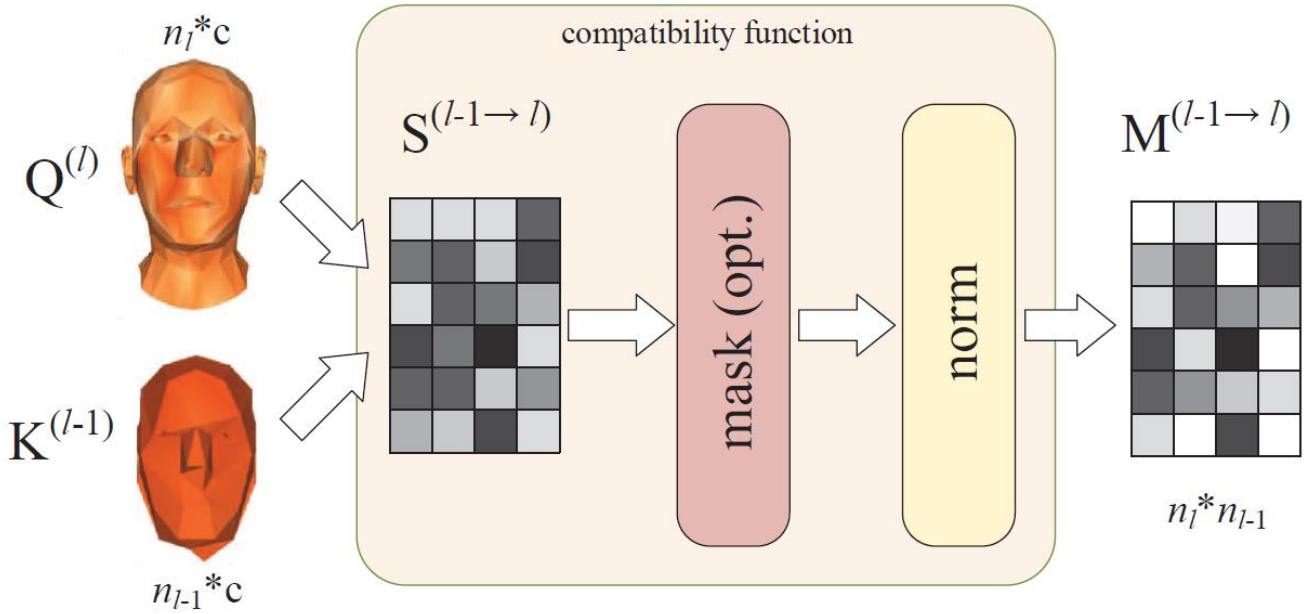
- attention
  - key
  - query

Method

- We propose an attention module to generate the mapping matrices, which allows to learn the mapping matrices end-to-end and use trainable keys and queries to compute the mapping matrices.
- By utilizing key and query, we can avoid over-parameterization of mapping matrix and exploit the non-local relationship between mesh vertices.

Method

- In Deep 3D Morphable Models, feature aggregation can be generally formulated as $$\boldsymbol{x}_i^{(l)} = \sum_{j=1}^{n_{l-1}} m_{ij}^{(l-1 \to l)} \boldsymbol{x}_j^{(l-1)}$$
- Feature aggregation via attention: $$m_{ij}^{(l-1 \to l)} = d\left(\boldsymbol{q}_i^{(l)}, \boldsymbol{k}_j^{(l-1)}\right)$$



Method

- Given the key and query vectors, the compatibility function measures how well two vertices at neighboring levels align:

$$s_{w,ij}^{(l-1\to l)} = \cos\left(\boldsymbol{q}_i^{(l)}, \boldsymbol{k}_j^{(l-1)}\right)$$

- Binary mask for the weight score:

$$b_{ij}^{(l-1\to l)} = \begin{cases} 1, & s_{w,ij}^{(l-1\to l)} \text{ is among the top } k \text{ of } s_{w,i:}^{(l-1\to l)} \\ 0, & \text{otherwise.} \end{cases}$$

- The fusion can be thought as a multi-head attention with a fixed head and a learnable head:

$$m_{ij}^{(l-1\to l)} = w_a m_{a,ij}^{(l-1\to l)} + (1 - w_a) m_{p,ij}^{(l-1\to l)}$$

Method

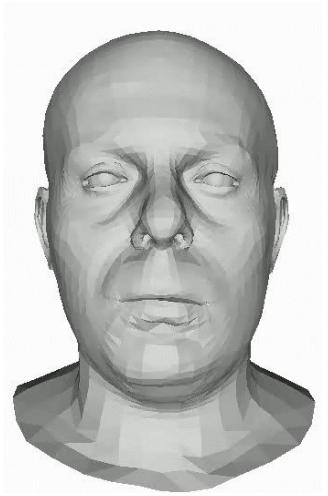
- We initialize the first three dimensions of the key and query vectors by the spatial position of vertices at the corresponding level. The remaining dimensions are randomly initialized by an uniform distribution.

Experiments

- We conduct experiments on human faces, bodies, and hands.

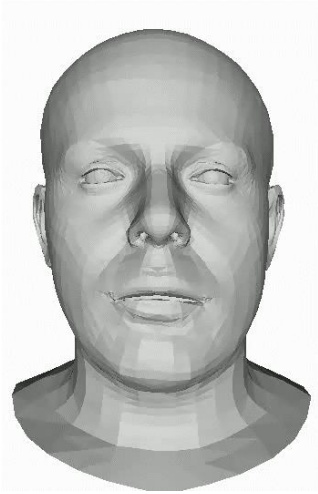
Mesh reconstruction

COMA
(Ranjan *et al.* 2018)

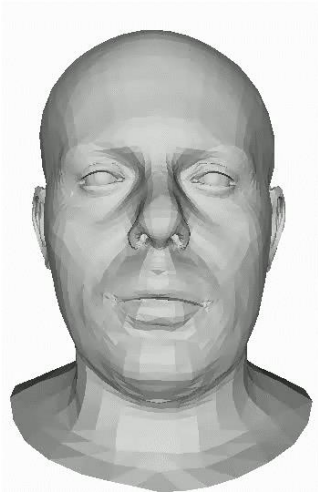


input

Ranjan *et al.*
2018

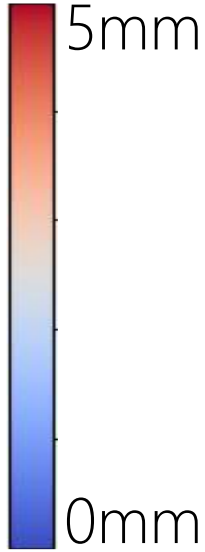


ours
Deep3DMM
(spectral)



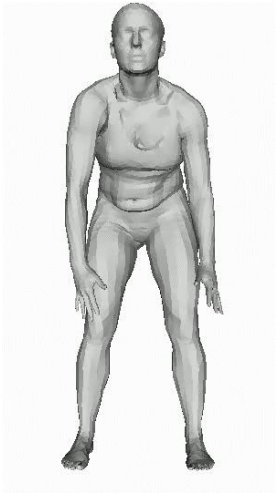
reconstruction

error



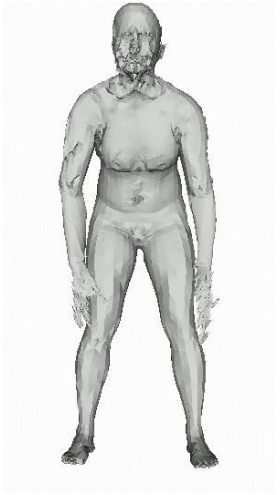
Mesh reconstruction

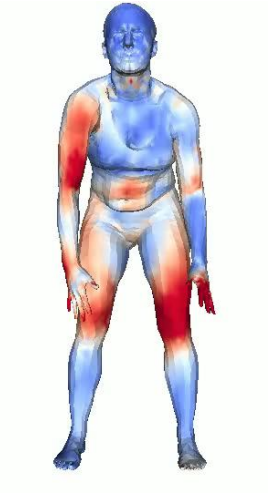
Dynamic FAUST
(Bogo *et al.* 2017)



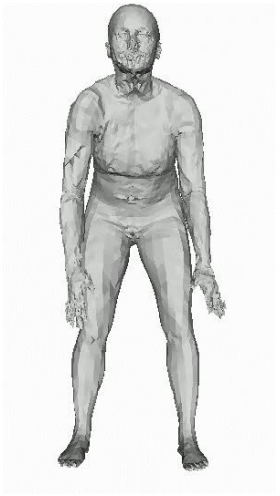
input

Ranjan *et al.*
2018



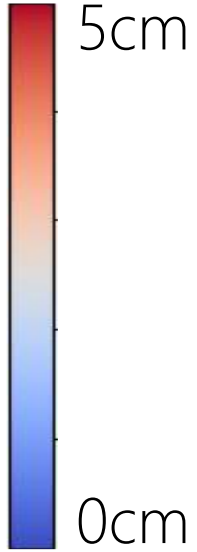


ours
Deep3DMM
(spectral)



reconstruction

error

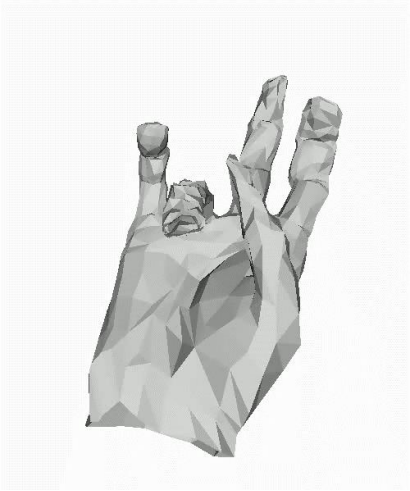


Mesh reconstruction

SynHand5M
(Malik *et al.* 2018)

input

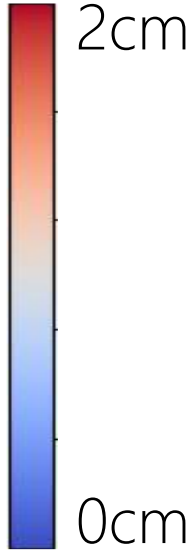
Ranjan *et al.*
2018



ours
Deep3DMM
(spectral)

reconstruction

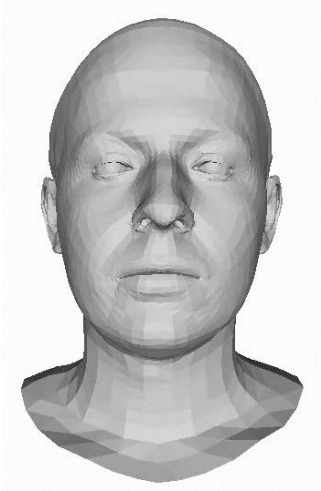
error

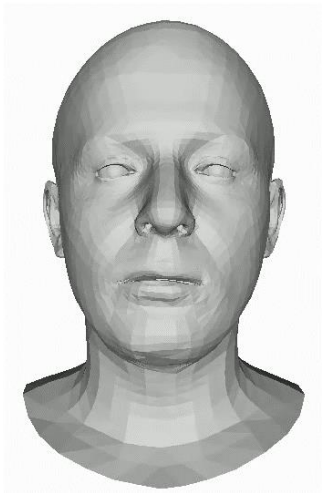


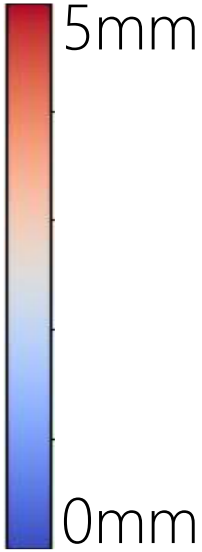
Mesh reconstruction

COMA
(Ranjan *et al.* 2018)

Bouritsas *et al.*
2019

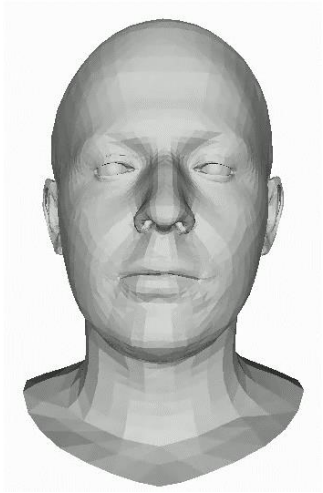






input

ours
Deep3DMM
(spiral)

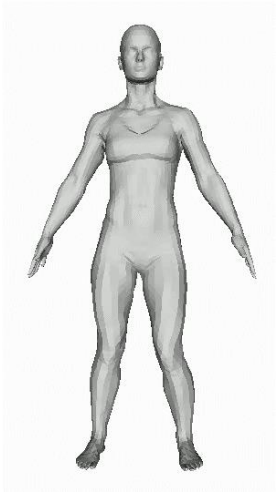


reconstruction

error

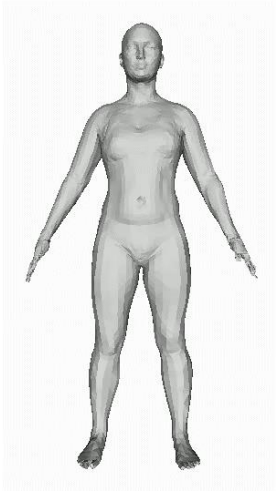
Mesh reconstruction

Dynamic FAUST
(Bogo *et al.* 2017)

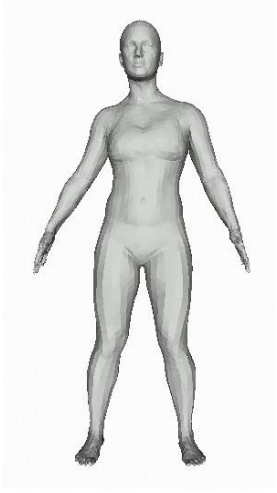


input

Bouritsas *et al.*
2019

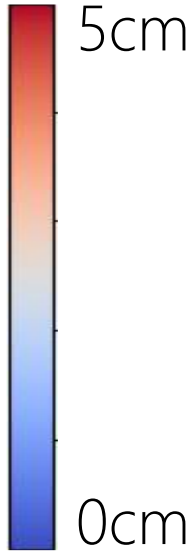


ours
Deep3DMM
(spiral)



reconstruction

error



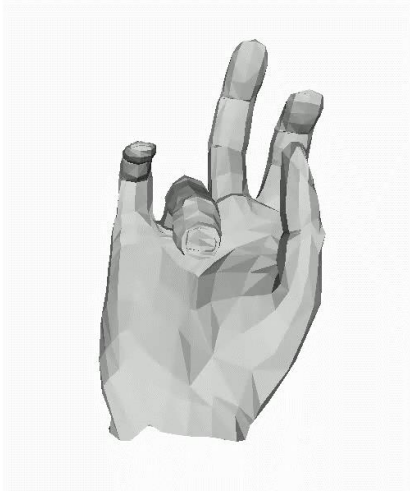
Mesh reconstruction

SynHand5M
(Malik *et al.* 2018)

input

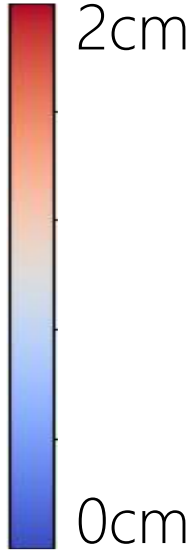
Bouritsas *et al.*
2019

ours
Deep3DMM
(spiral)



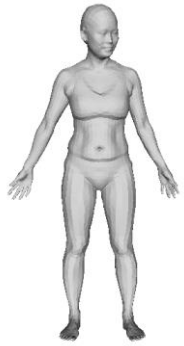
reconstruction

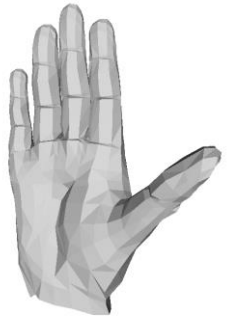
error

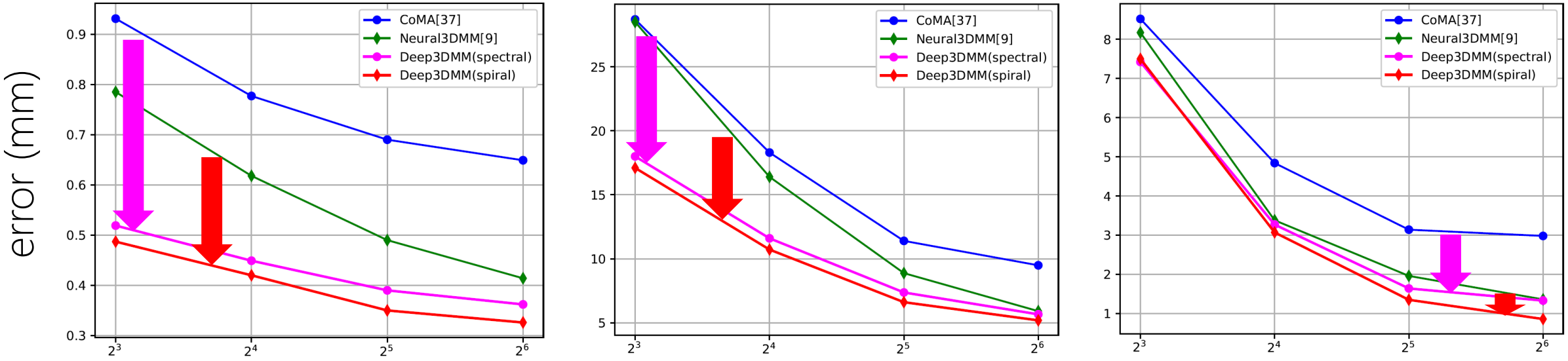


Mesh reconstruction

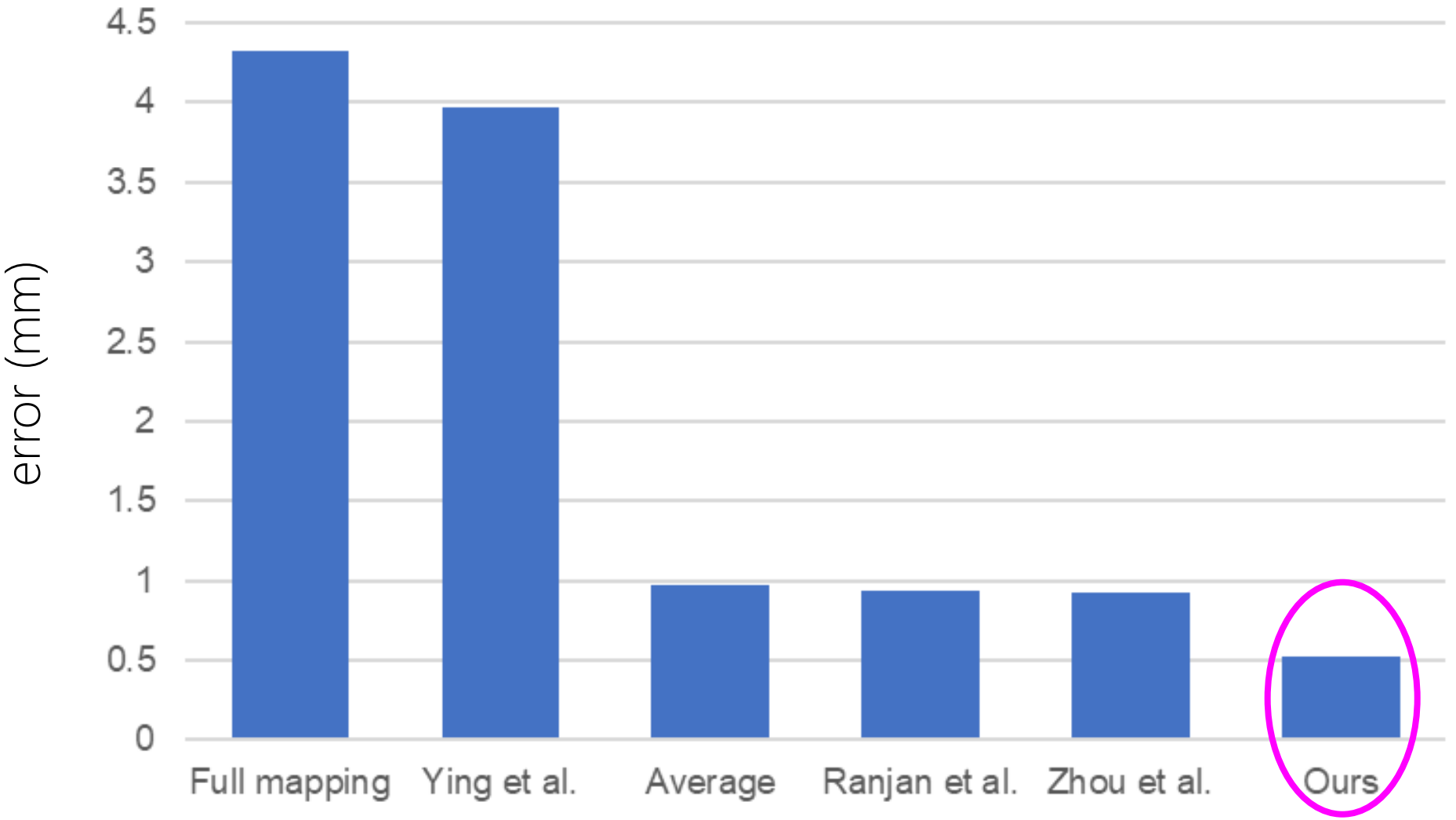
statistical reconstruction errors for different latent dimensions: 8, 16, 32, 64







Comparisons to existing aggregation methods.



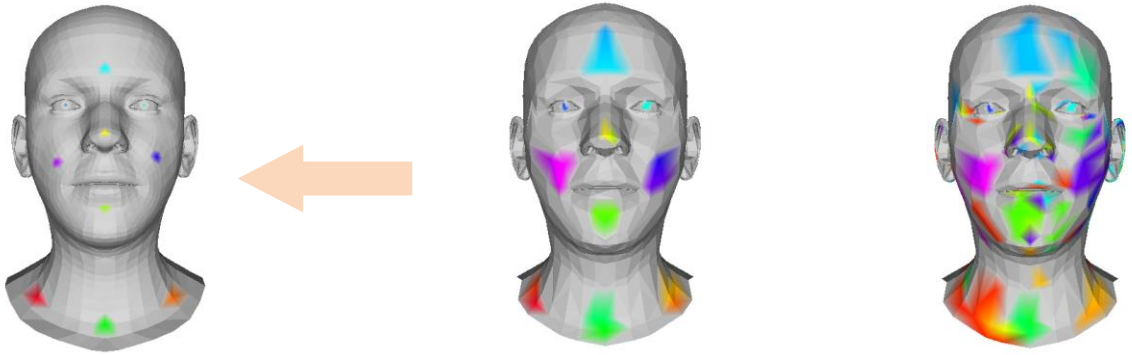
COMA (Ranjan *et al.* 2018)

Visualization of mapping matrices: Receptive fields

Ranjan *et al.* 2018

ours

COMA
(Ranjan *et al.* 2018)

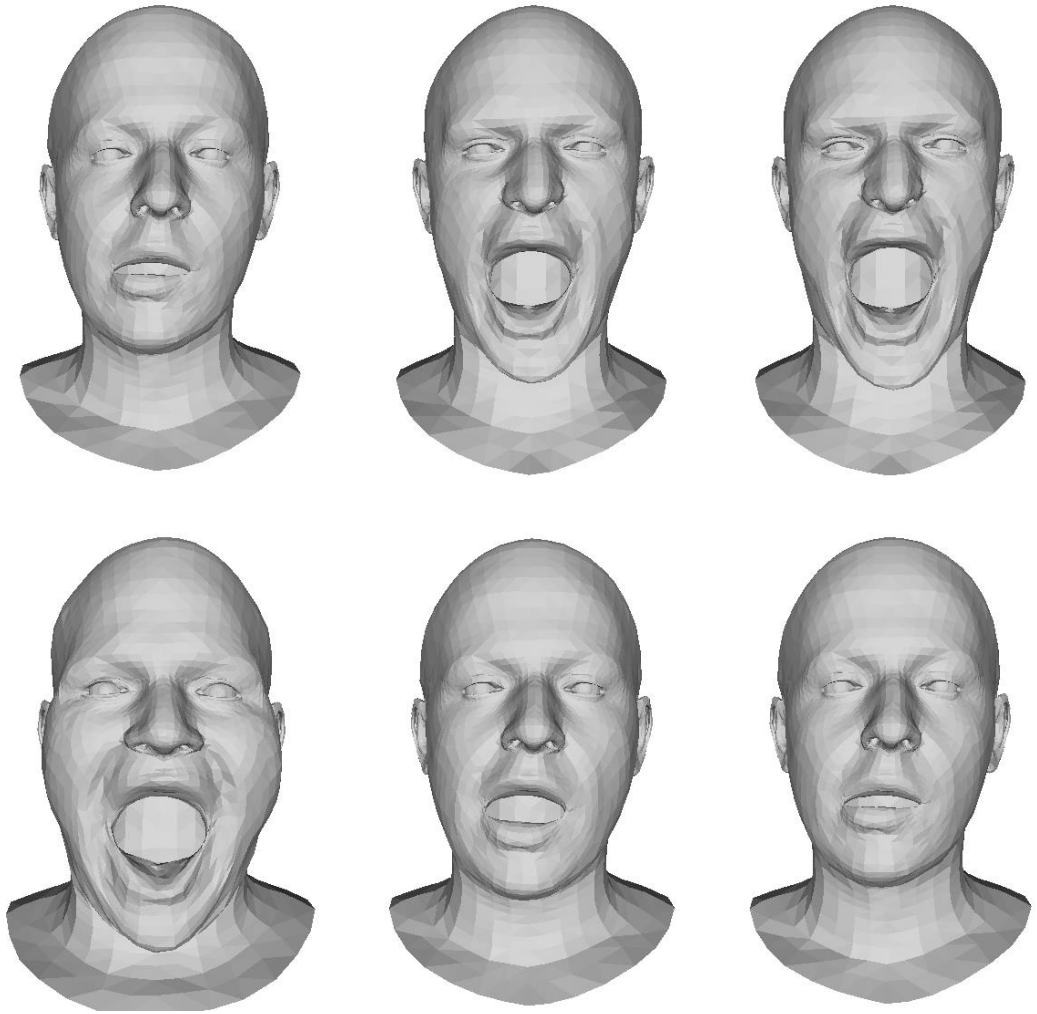


Dynamic FAUST
(Bogo *et al.* 2017)

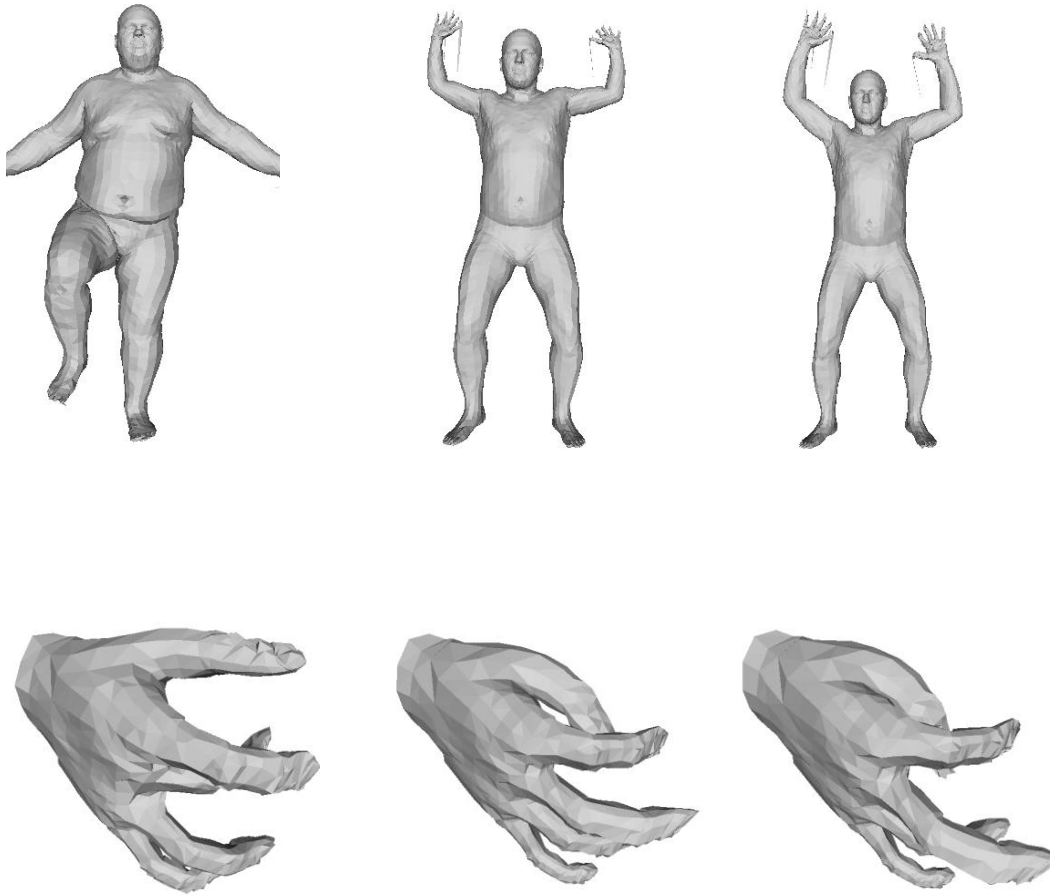
epoch=350
epoch=90

SynHand5M
(Malik *et al.* 2018)

Interpolation in latent space

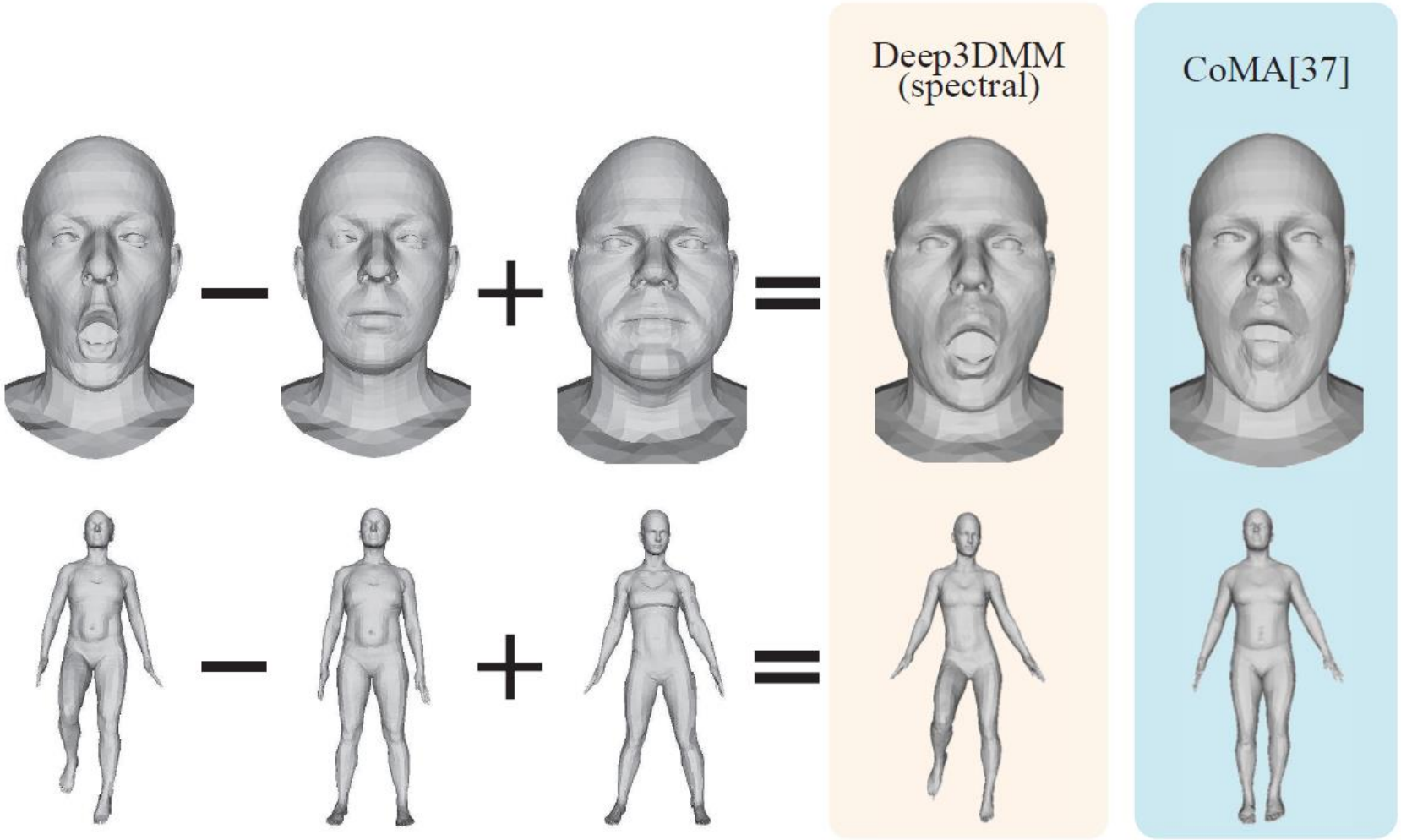


source interpolation target



source interpolation target

Deformation transfer with arithmetic operations on latent representations



Conclusion

- We propose to learn feature aggregation for deep 3DMMs.
- Our attention based feature aggregation uses trainable keys and queries to compute the mapping matrices, such that the matrices can be optimized by the target objective and used as a train stage only drop-in replacement for either down-sampling or up-sampling.
- Experimental results show the learned aggregation can enhance the capacity of deep 3DMMs with isotropic or anisotropic convolutions.

- Code: https://github.com/zxchen110/Deep3DMM

- [1] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3D faces using convolutional mesh autoencoders, ECCV 2018.

CVPR VIRTUAL JUNE 19-25

HUAWEI FACER2VM KAIST 1971 Imperial College London

Sequential Graph Convolutional Network for Active Learning

Razvan Caramalau[1], Binod Bhattarai[1], Tae-Kyun Kim[1,2]



[1]Imperial College London, UK
[2]KAIST, South Korea

Motivation

- Active Learning (AL) – a critical area of research in data-hungry deep learning networks for vision applications and much more.

- The annotation process for large-scale datasets is time-consuming, expensive, needs experts, and most-times noisy. Active Learning overcomes this by reducing the labelled set while maintaining the most meaningful examples.

- New model-based AL approaches exploit the correlation between the labelled and the unlabelled images while inheriting the learner's uncertainty.

Recent works

Model-base Active Learning:

- Learning Loss(CVPR 2019) introduces a separate loss-prediction module to be trained together with the learner.

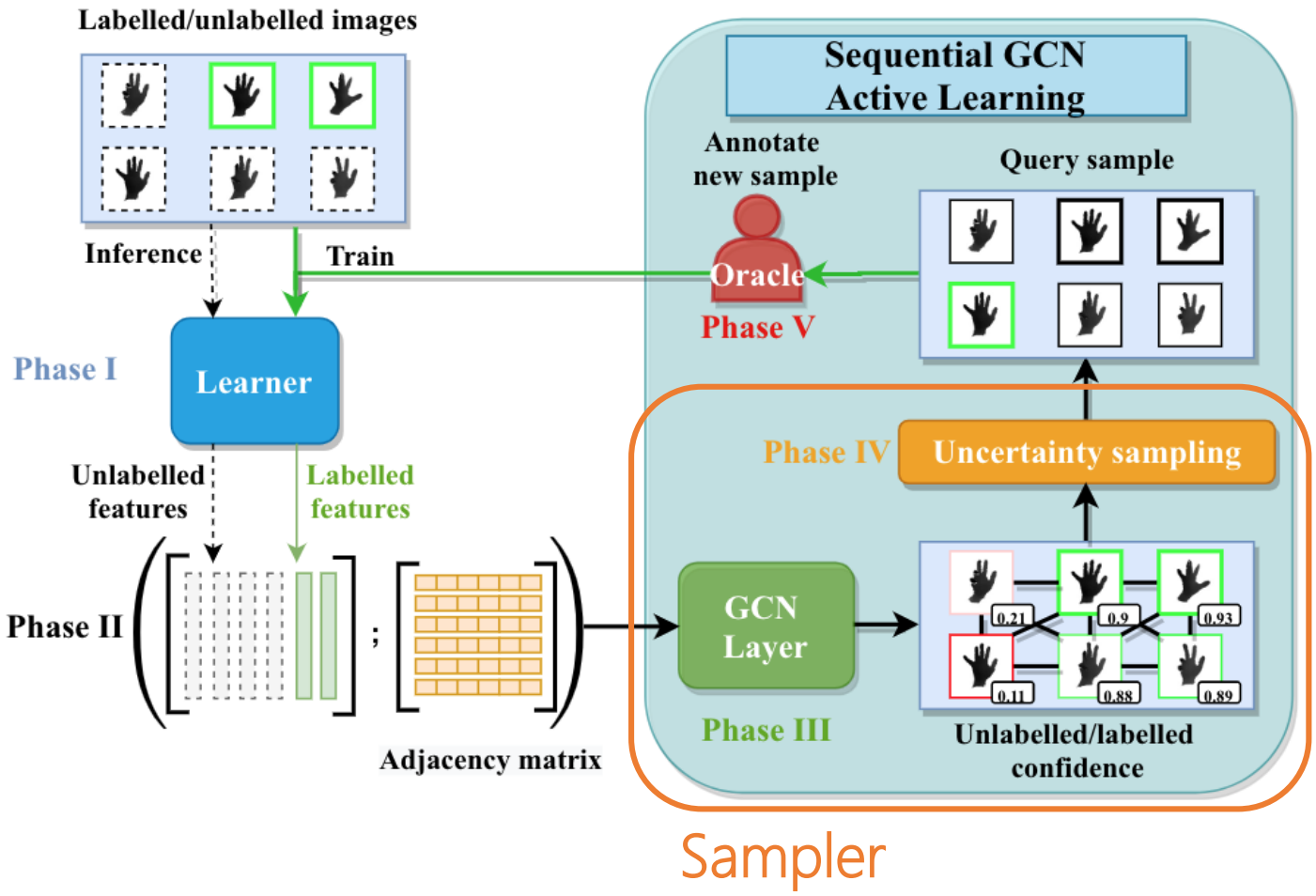
- VAAL(ICCV 2019) trains a variational auto-encoder (VAE) that learns a latent space for better discrimination between labelled and unlabelled images in an adversarial manner.

- Both lack of a mechanism that exploits the correlation between the labelled and unlabelled images

=> Graph Convolutional Networks(GCNs) are capable of sharing information between the nodes via message-passing operations

Proposed pipeline

- **Phase I** implements the learner. This is a model trained to minimize the objective of the downstream task.
- **Phase II, III and IV** compose our sampler where we deploy the GCN and apply the sampling techniques on graph-induced node embeddings and their confidence scores.
- **Phase V**, the selected unlabelled examples are sent for annotation.



Methods - Learner

- Initially, the learner is trained with a small number of seed labelled examples. We extract the features of both labelled and unlabelled images from the learner parameters.
- Classification: We took ResNet-18 as the CNN model. The objective of our classifier is cross entropy:

$$\mathcal{L}^{c}_{\mathcal{M}}(\mathbf{x}, \mathbf{y}; \theta) = -\frac{1}{N_l} \sum_{i=1}^{N_l} \mathbf{y}_i \log(f(\mathbf{x}_i, \mathbf{y}_i; \theta))$$

- Regression: To tackle the 3D HPE, we deploy *DeepPrior* [26] architecture. J is the number of joints of hand pose:

$$\mathcal{L}^{r}_{\mathcal{M}}(\mathbf{x}, \mathbf{y}; \theta) = \frac{1}{N_l} \sum_{i=1}^{N_l} \left(\frac{1}{J} \sum_{j=1}^{J} \|\mathbf{y}_{i,j} - f(\mathbf{x}_{i,j}, \mathbf{y}_{i,j}; \theta)\|^2 \right)$$

Methods - Sampler

- A pool-based scenario for active learning:

$$\min_{n} \min_{\mathcal{L}_{\mathcal{M}}} \mathcal{A}(\mathcal{L}_{\mathcal{M}}(\mathbf{x}, \mathbf{y}; \theta) | \mathbf{D}_0 \subset \cdots \subset \mathbf{D}_n \subset \mathbf{D}_U)$$

We aim to minimise the number n of active learning stages so that fewer samples (x, y) would require annotation.

- During Phase II, we construct a graph where features are used to initialise the nodes of the graph and similarities represent the edges.
- The features extracted from the learner creates an opportunity to inherit uncertainties to the sampler.

Methods - Graph Convolutional Network

- This graph is passed through GCN layers (Phase III) and the parameters of the graph are learned to identify the nodes of labelled vs unlabelled examples. We convolve on the graph which does message-passing operations between the nodes to induce the higher-order representations.
- To avoid over-smoothing of the features in GCN [18], we adopt a two-layer architecture.

$$f_{\mathcal{G}} = \sigma(\Theta_2(ReLU(\Theta_1 A)A)$$

- The loss for GCN:

$$\mathcal{L}_{\mathcal{G}}(\mathcal{V}, A; \Theta_1, \Theta_2) = -\frac{1}{N_l}\sum_{i=1}^{N_l}\log(f_{\mathcal{G}}(\mathcal{V}, A; \Theta_1, \Theta_2)_i) - \frac{\lambda}{N - N_l}\sum_{i=N_l+1}^{N}\log(1 - f_{\mathcal{G}}(\mathcal{V}, A; \Theta_1, \Theta_2)_i)$$

- The graph embedding of any image depends primarily upon the initial representation and the associated neighbourhood nodes.
- Thus, the images bearing similar semantic and neighbourhood structure end up inducing close representations which will play a key role in identifying the sufficiently different unlabelled examples from the labelled ones.
- The nodes after convolutions are classified as labelled or unlabelled.

Methods

Uncertainty sampling on GCN

- While querying a fixed number of b points for a new subset, we apply the following equation:

$$\mathbf{D}_L = \mathbf{D}_L \cup \underset{i=1\cdots b}{\arg\max} |s_{margin} - f_{\mathcal{G}}(\mathbf{v}_i; \mathbf{D}_U)|$$

- For selecting the most uncertain unlabelled samples, margin should be closer to 0.

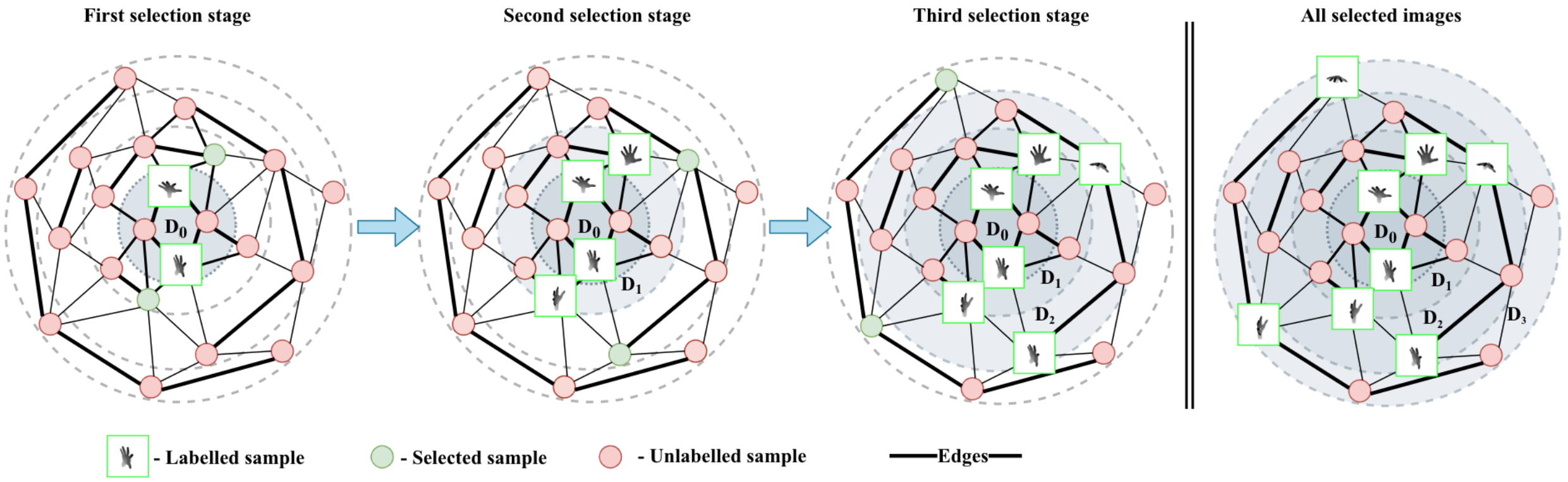
CoreSet sampling on GCN

- To integrate geometric information between the labelled and unlabelled graph representation, we approach a CoreSet technique [31]:

$$\mathbf{D}_L = \mathbf{D}_L \cup \underset{i \in \mathbf{D}_U}{\arg\max} \underset{j \in \mathbf{D}_L}{\min} \delta(f^1_{\mathcal{G}}(A, \mathbf{v}_i; \Theta_1), f^1_{\mathcal{G}}(A, \mathbf{v}_j; \Theta_1))$$

where δ is the Euclidean distance.

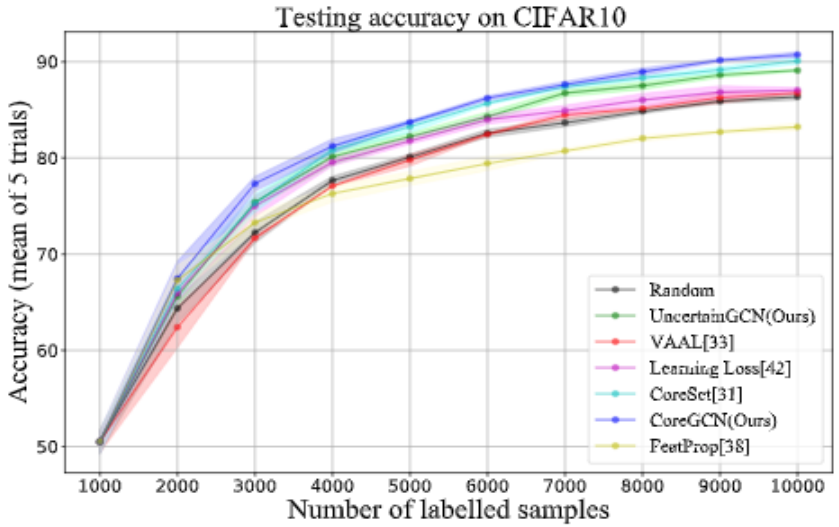
UncertainGCN – representative simulation

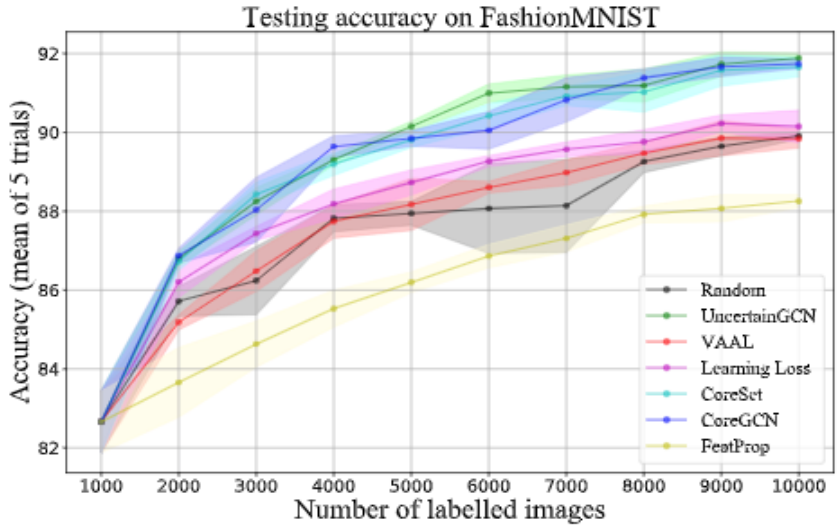


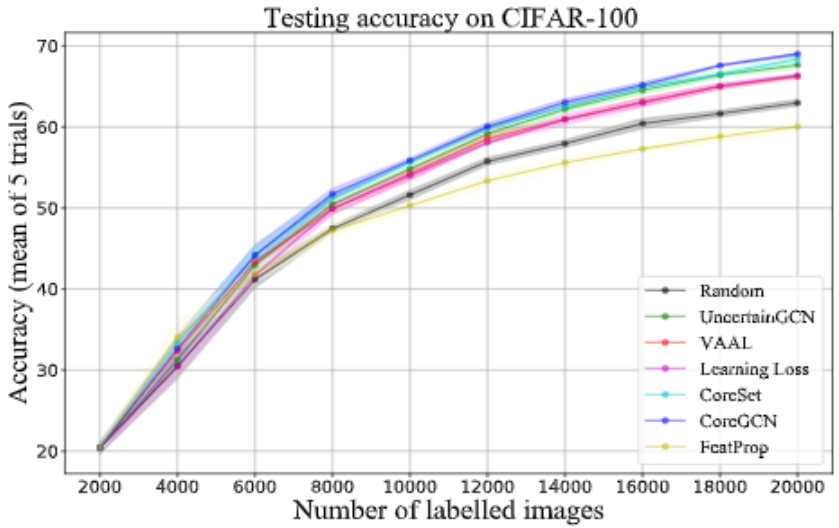
Experiments

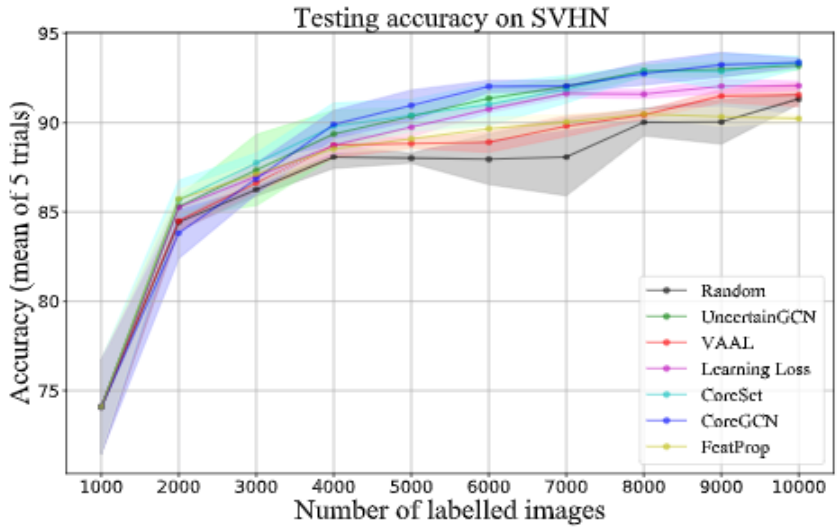
- In the experiment section, we compare several baselines on image classification and 3D hand pose estimation datasets. Because these benchmarks are fully labelled, we can empirically evaluate under different budget and labelled/unlabelled subset settings (more details in the main paper).
- We achieve state-of-the-art testing performance in challenging applications.

Image Classification Quantitative comparison







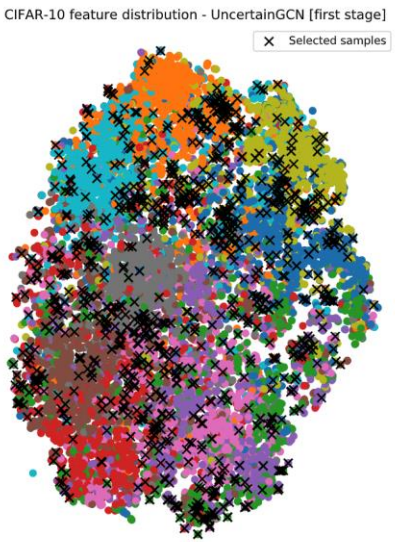


Qualitative exploration

First Active learning selection cycle

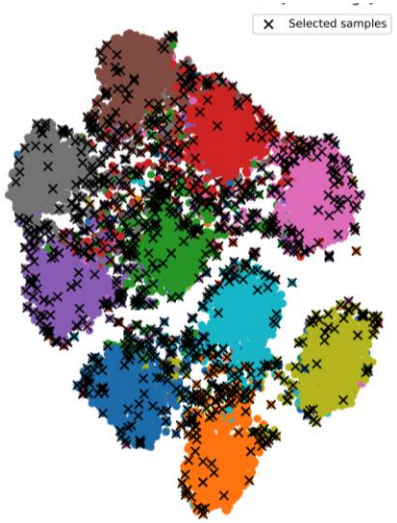
CoreSet

UncertainGCN



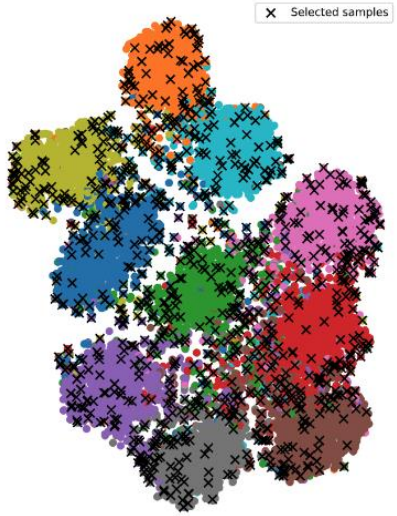
Fourth Active learning selection cycle

CoreSet

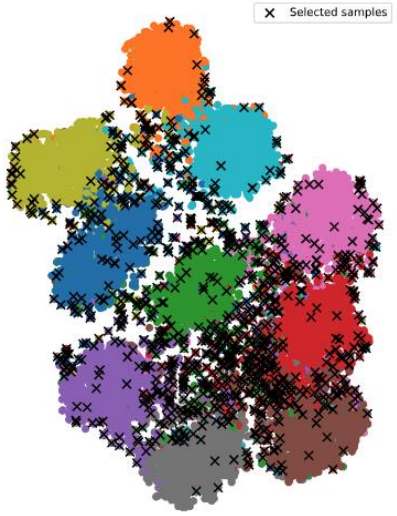


UncertainGCN

CoreSet

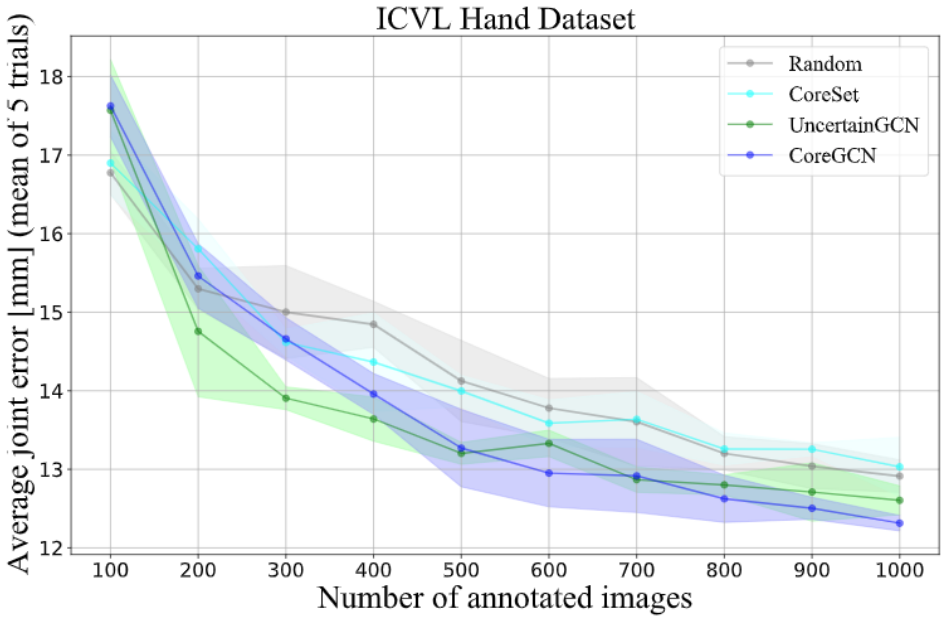


CoreGCN

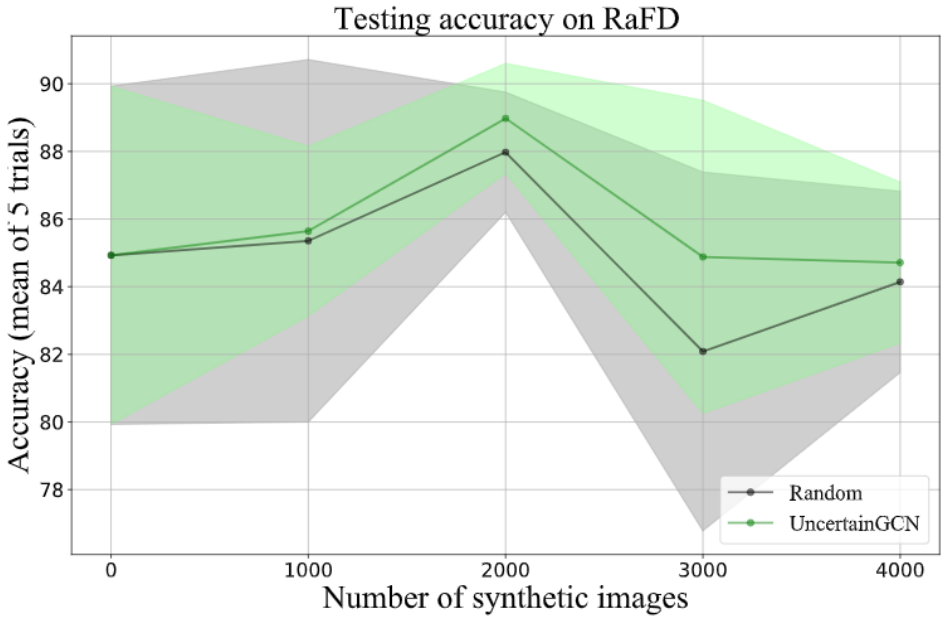


Other datasets: ICVL 3D Hand Pose Estimation – RaFD Face Expression augmented data

ICVL 3D Hand Pose Estimation Dataset



RaFD - StarGAN Face Synthetic Dataset



Ablation studies

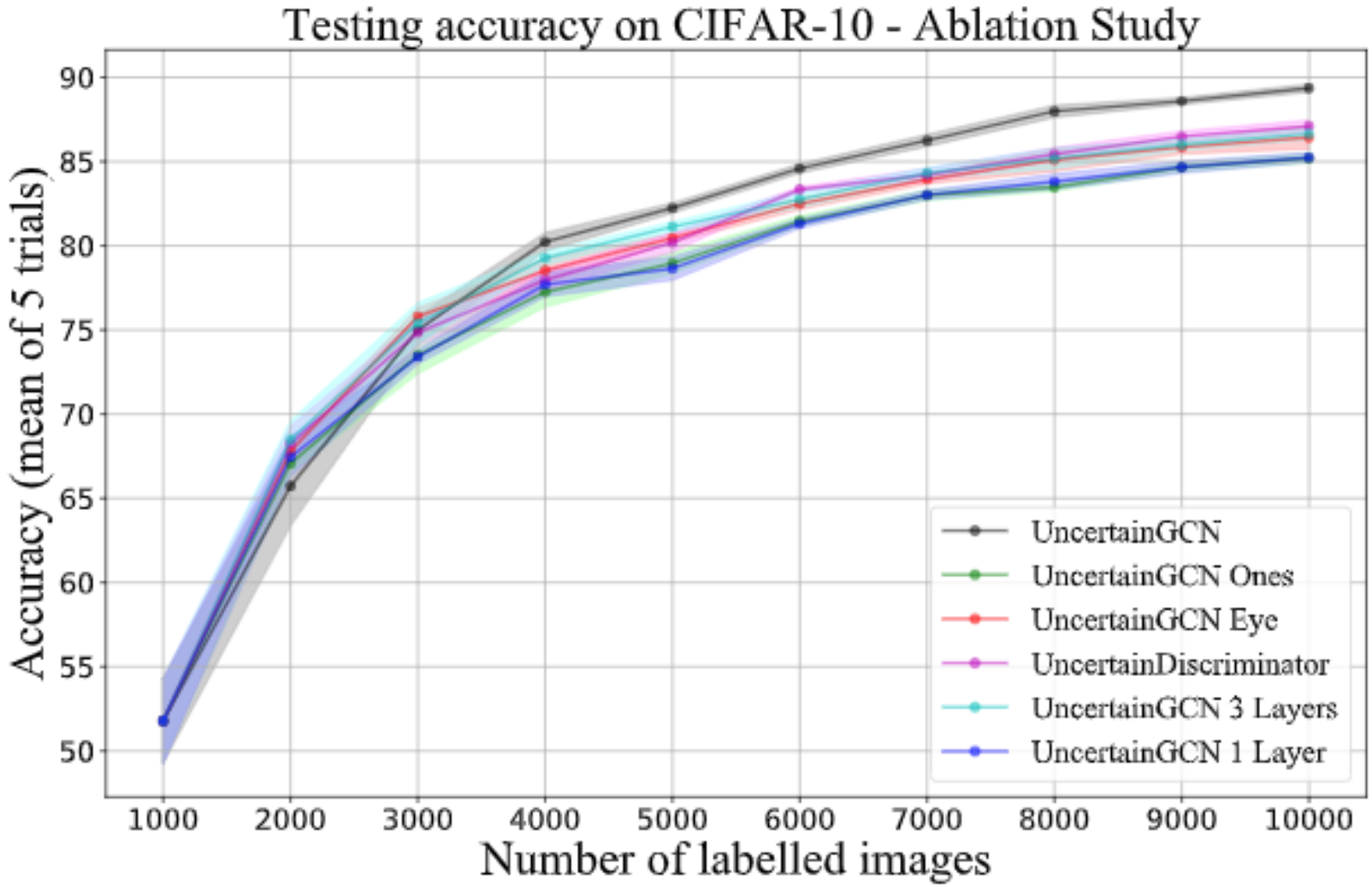
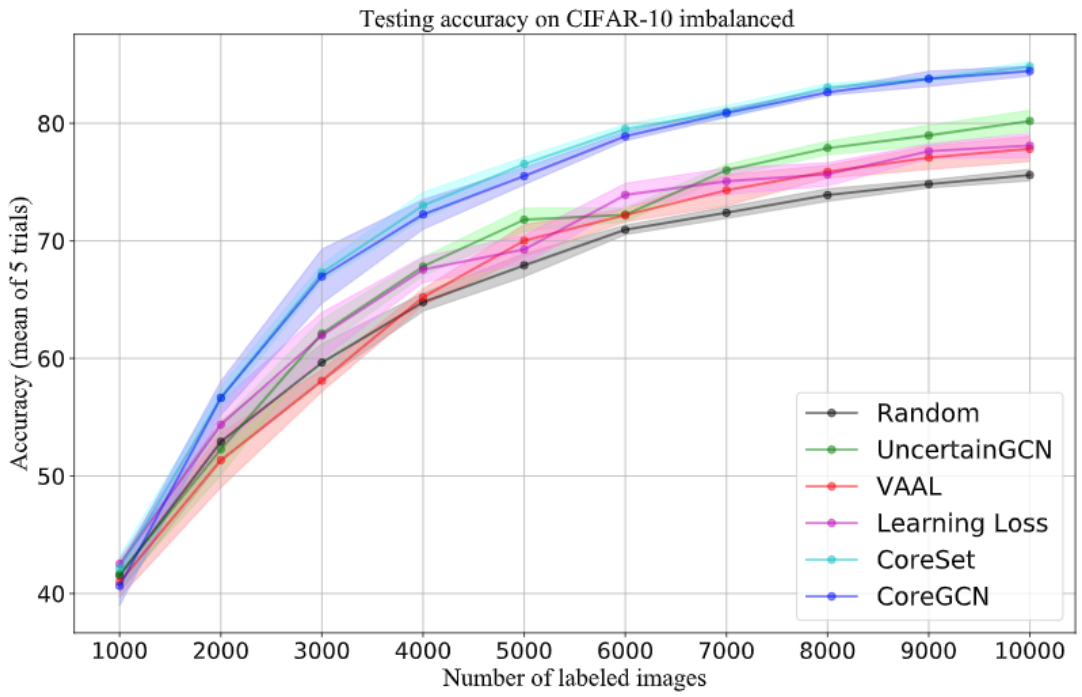


Image classification – special scenarios

CIFAR-10 imbalanced classes scenario



CIFAR-10 VGG-11 learner scenario

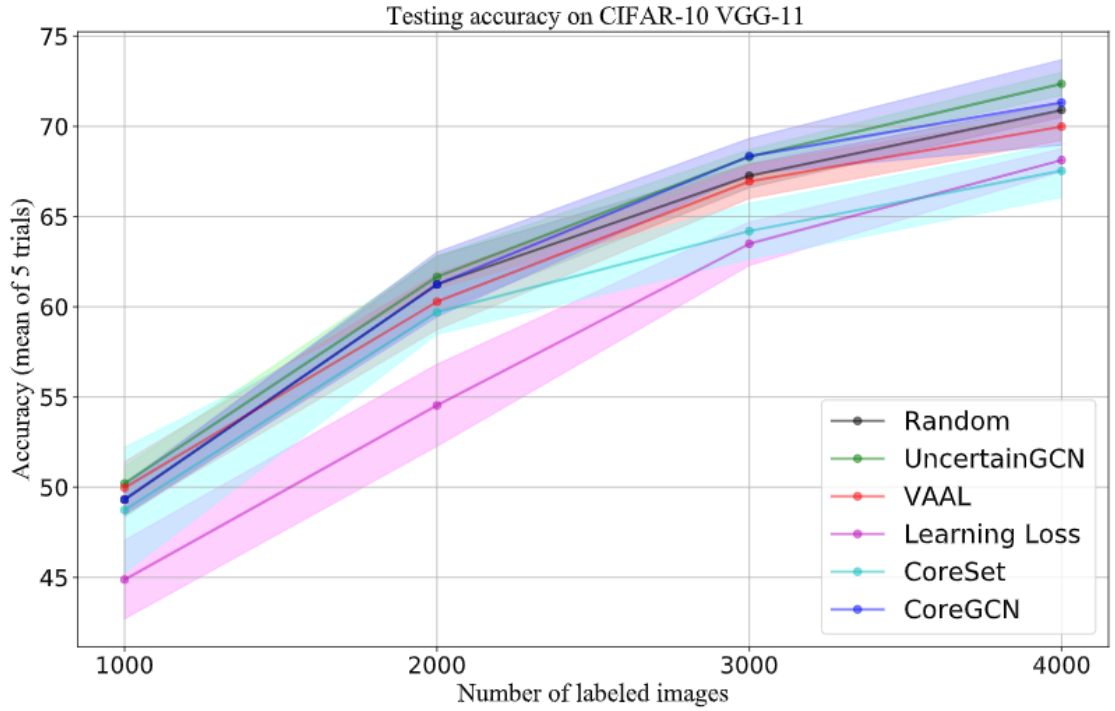
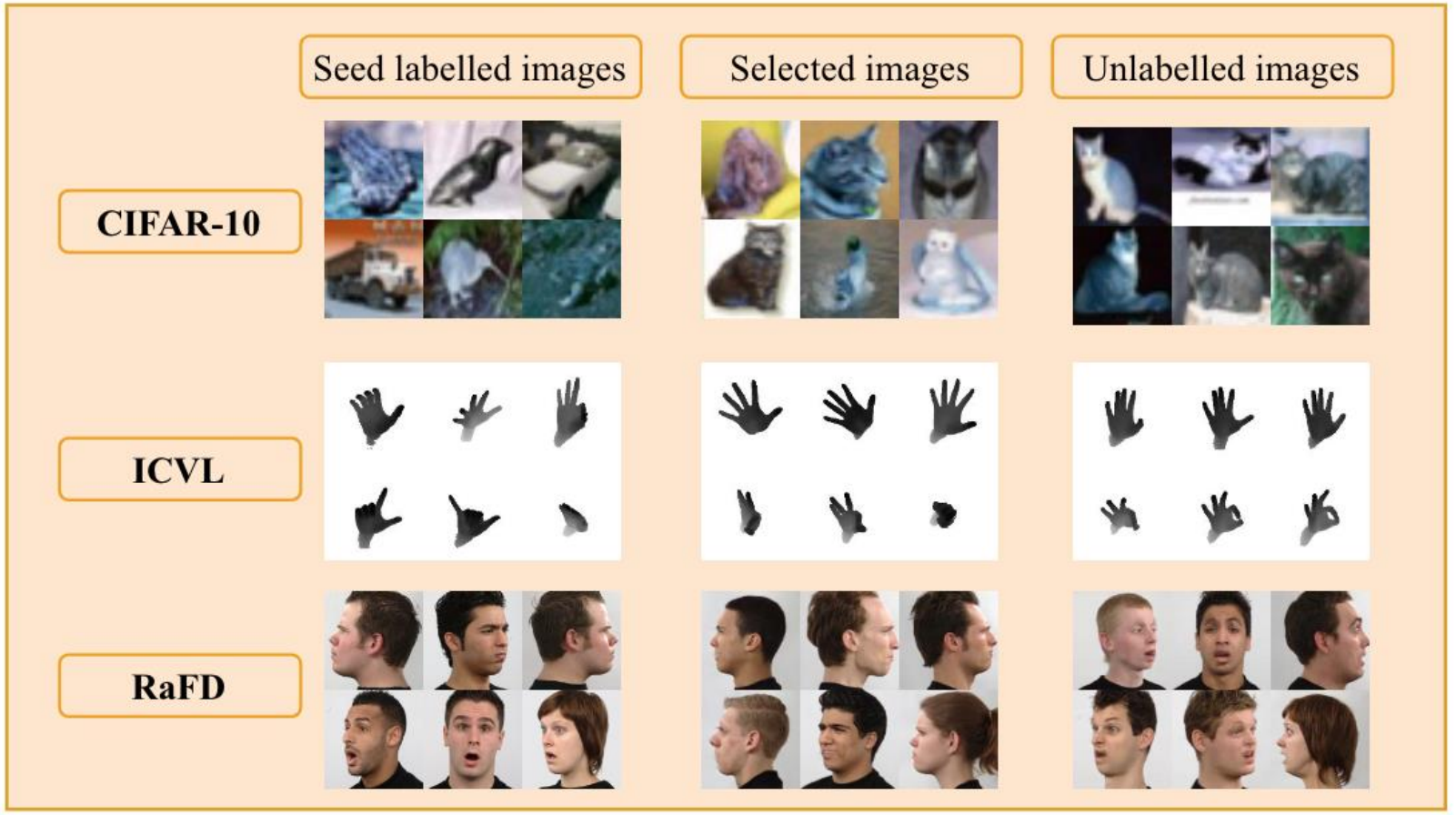


Image examples at the last Active Learning selection cycle



Things to take home:

- A novel methodology of active learning in image classification and regression using Graph Convolutional Network.

- Our sampling techniques, **UncertainGCN** and **CoreGCN**, produced state-of-the-art results on 6 benchmarks and in limited scenarios.

- These methods **maximise informativeness** within the data space while allowing integration into other learning tasks.

FACER2VM

Thank you for the interest in our work!

This work is partially supported by Huawei Technologies Co. and by EPSRC Programme Grant FACER2VM.