# DEMEA: Deep Mesh Autoencoders for Non-Rigidly Deforming Objects

Edgar Tretschk Ayush Tewari
Vladislav Golyanik Michael Zollhöfer Christian Theobalt

Graphics Vision & Video
max planck institut informatik
SIC Saarland Informatics Campus
Stanford University

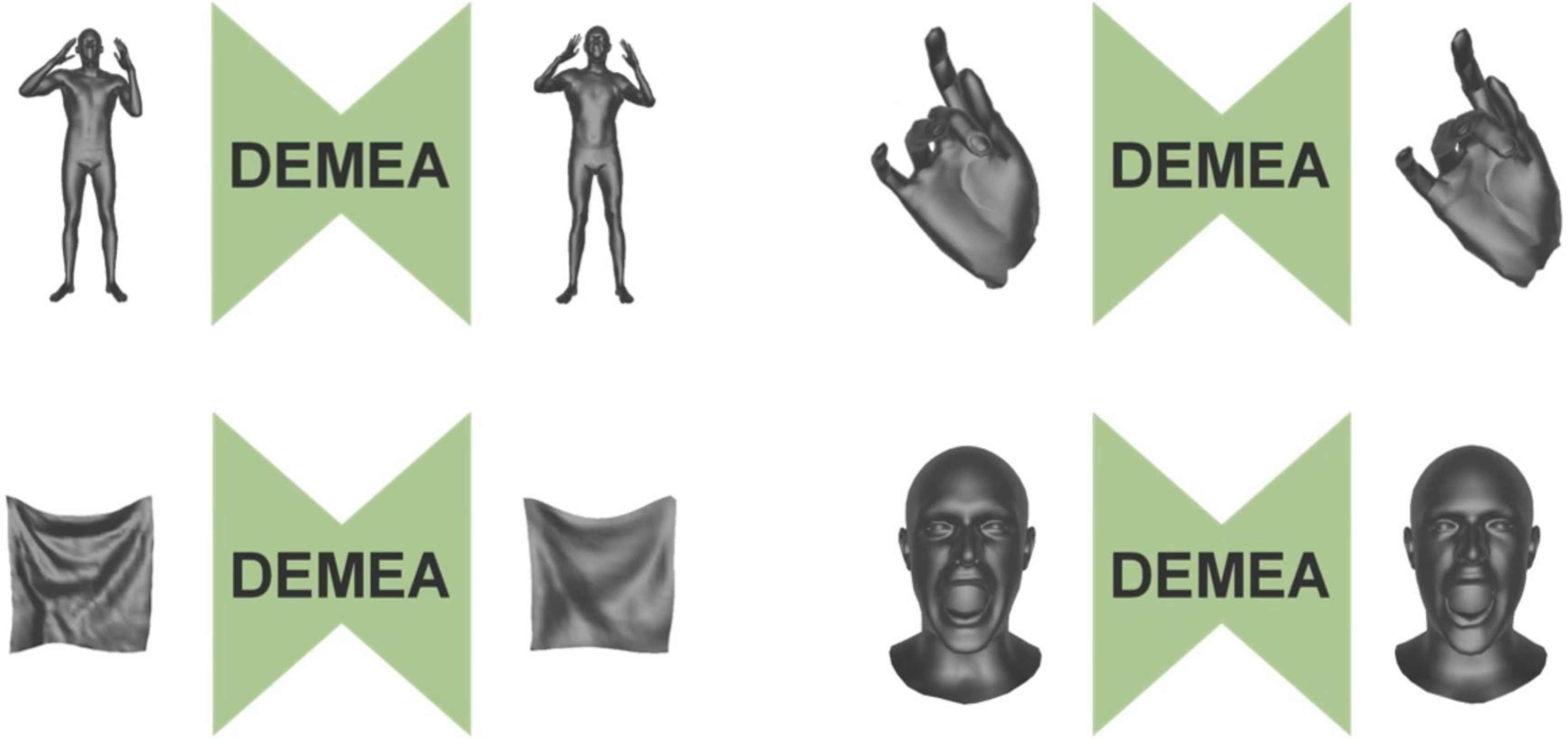
ECCV'20 ONLINE 23-28 AUGUST 2020

# Introduction

Goal: dimensionality reduction of mesh data

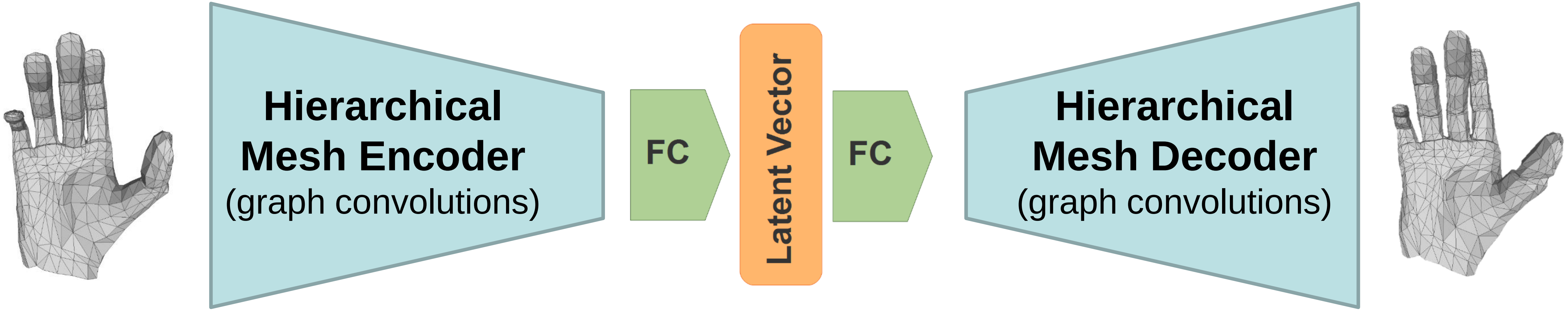
- Focus on general non-rigid objects

→ Mesh autoencoding

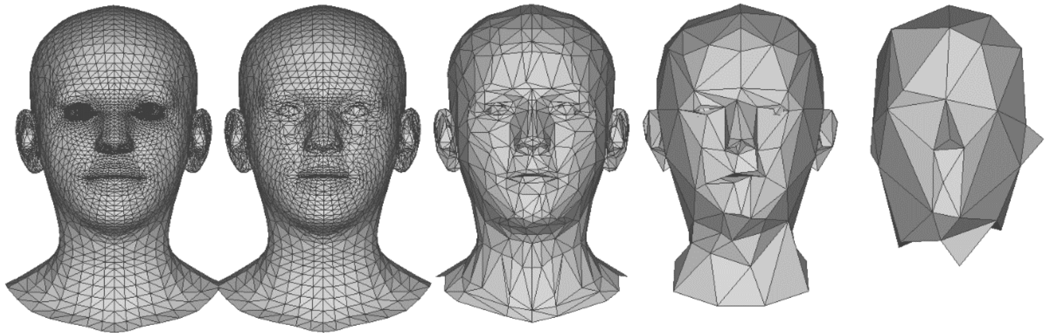


# Related Work

- CoMA: Convolutional Mesh Autoencoder (Ranjan et al. 2018)

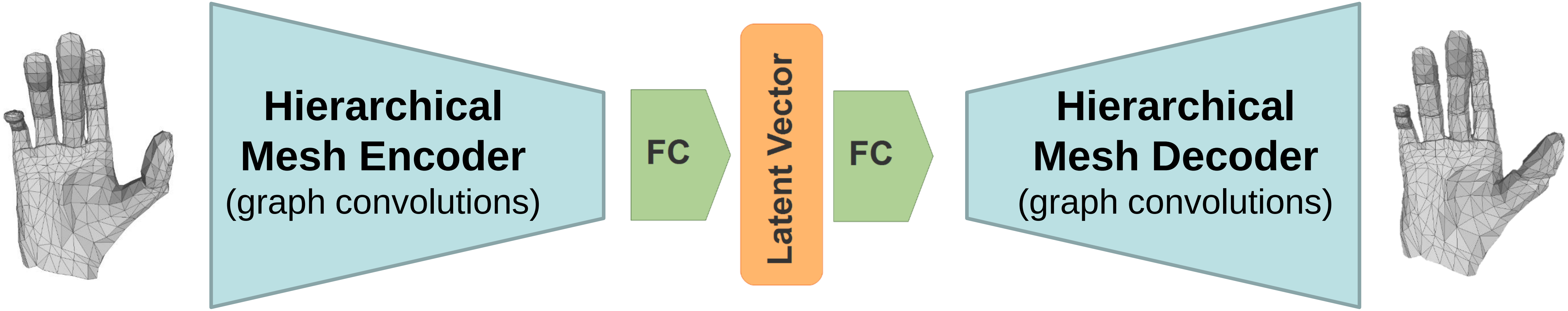



  - Operations on each hierarchy level:
    - Chebyshev graph convolution (Defferrard et al. 2016)
    - Downsampling (by factor 4)

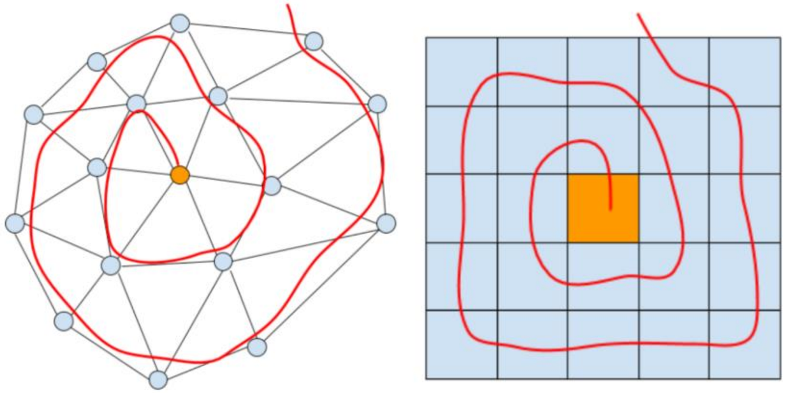


# Related Work

- Neural 3DMM: Neural 3D Morphable Model (Bouritsas et al. 2019)

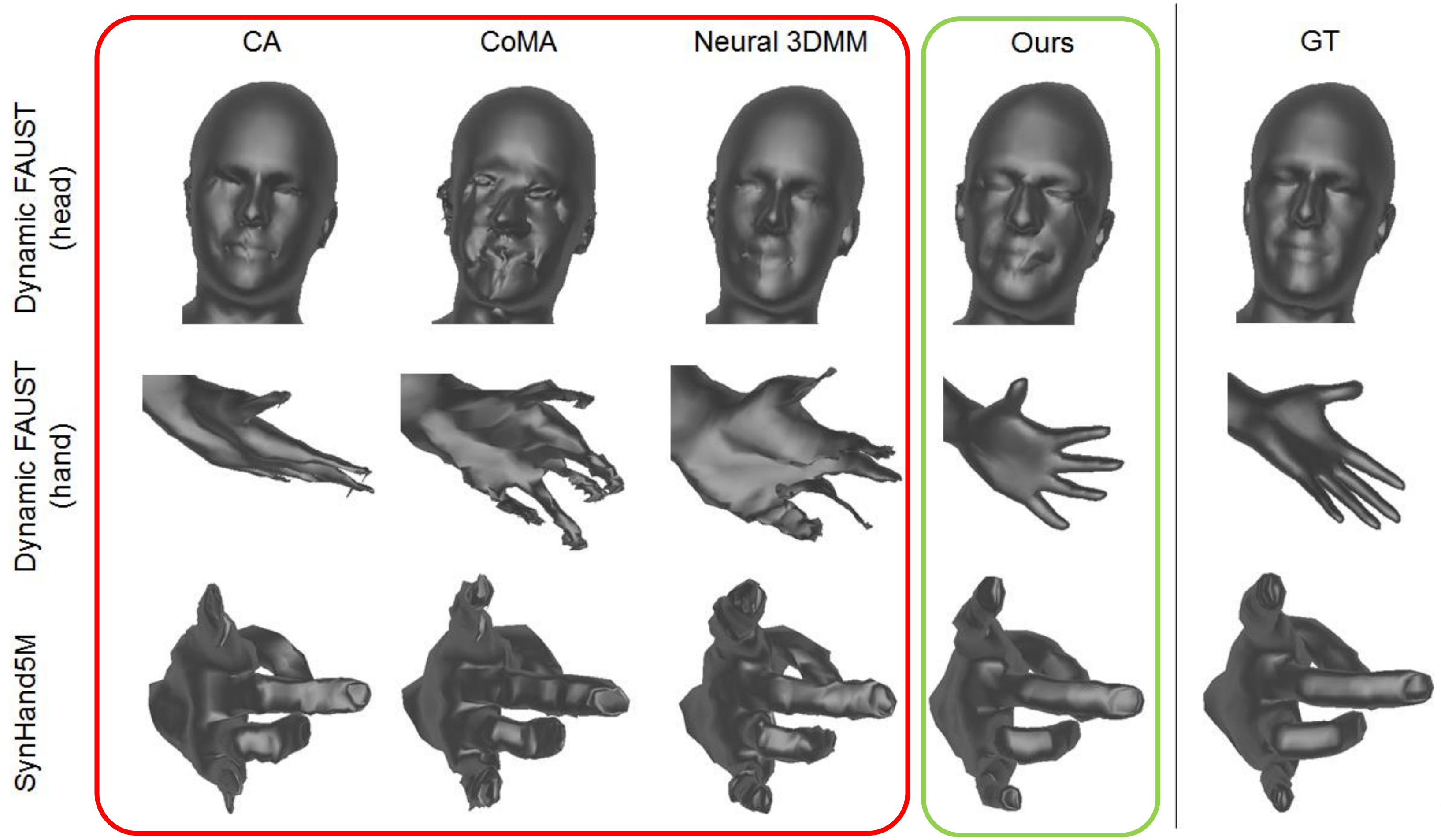



  - Operations on each hierarchy level:
    - **Spiral graph convolution**
    - Downsampling (by factor 4)



# Motivation

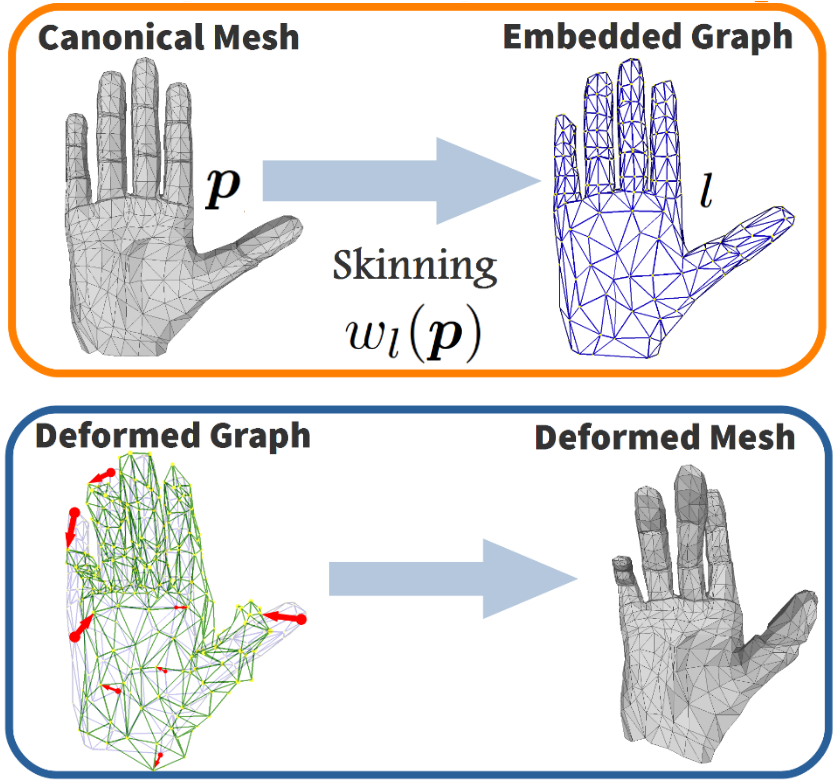
- Previous work does not explicitly model non-rigid deformations
  → DEMEA does!




Dynamic FAUST: Bogo et al. 2017
SynHand5M: Malik et al. 2018

# Background

- General non-rigid deformation priors from computer graphics, e.g.:
  - As-Rigid-As-Possible Deformation (Sorkine et al. 2007)
  - **Embedded Deformation (Sumner et al. 2007)**




Deformed position **G**(***p***) of vertex ***p*** is a linear combination:

$$\mathbf{G}(\boldsymbol{p}) = \sum_{l \in \mathcal{N}_{\boldsymbol{p}}} w_l(\boldsymbol{p}) \cdot \boldsymbol{T}_l(\boldsymbol{p})$$

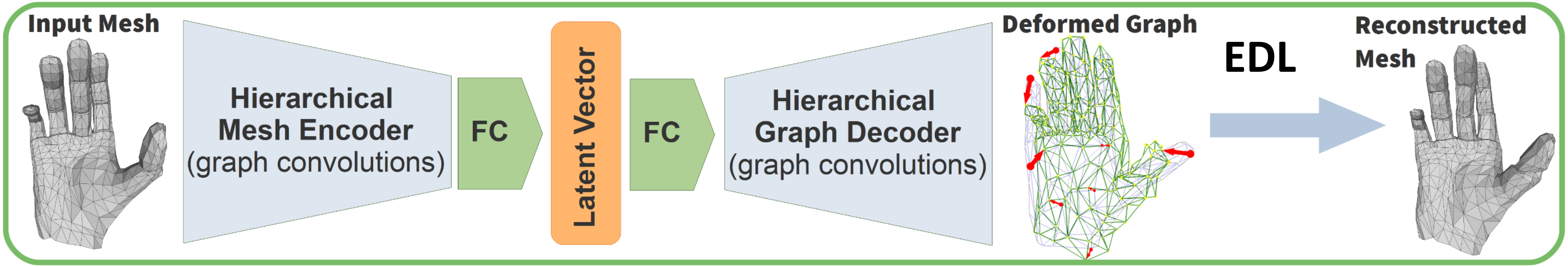
Each graph node ***l*** induces its own affine transformation around its canonical position $\boldsymbol{g}_l$:

$$\boldsymbol{T}_l(\boldsymbol{p}) = \boldsymbol{R}_l[\boldsymbol{p} - \boldsymbol{g}_l] + \boldsymbol{g}_l + \boldsymbol{t}_l$$

The affine transform is parametrized by a rotation $\boldsymbol{R}_l$ and a translation $\boldsymbol{t}_l$

# Method

- Combine learning-based dimensionality reduction with model-based **embedded deformation layer (EDL):**



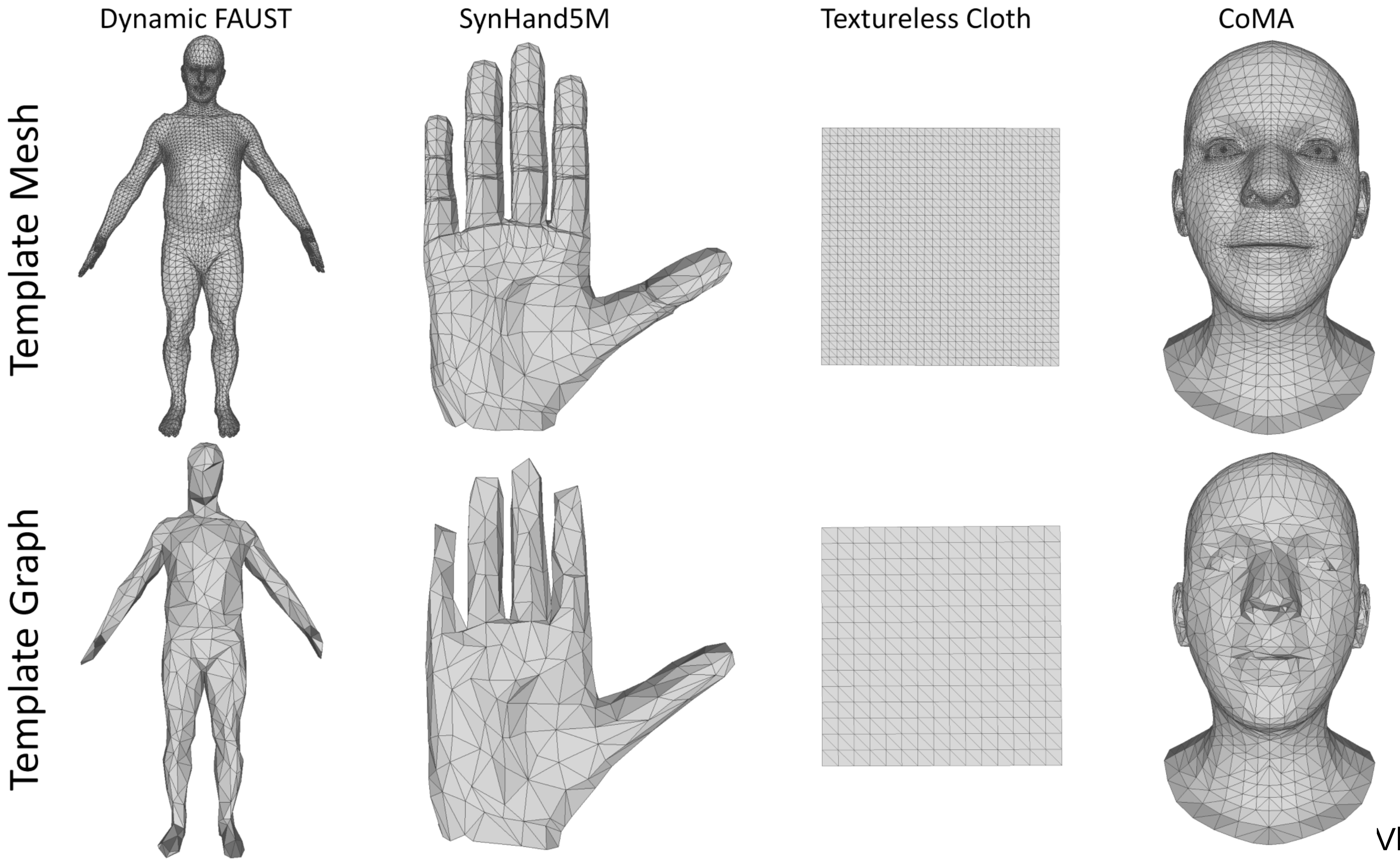

  - Last layer of graph decoder regresses rotation and translation of each graph node
    - Physically interpretable intermediate representation
    - Apply EDL to obtain full mesh
      - EDL is differentiable → can be used for backpropagation

# Data

- Four datasets of non-rigid objects:
  - Humans: Dynamic FAUST (Bogo et al. 2017)
  - Hands: SynHand5M (Malik et al. 2018)
  - Faces: CoMA (Ranjan et al. 2018)
  - Cloth: Textureless Cloth (Bednarik et al. 2018)
- Like prior work, we train category-specific auto-encoders

# Data: Embedded Graphs

- EDL needs embedded graphs:



# Results: Baselines

- CA/MCA: like DEMEA but EDL is replaced with graph convolutions
- FCA: encoder and decoder are a single FC each
- FCED: like FCA but EDL is added

| | DFaust | | SynHand5M | | Cloth | | CoMA | |
|---|---|---|---|---|---|---|---|---|
| | 8 | 32 | 8 | 32 | 8 | 32 | 8 | 32 |
| CA | 6.35 | **2.07** | 8.12 | 2.60 | **11.21** | 6.50 | **1.17** | 0.72 |
| MCA | **6.21** | 2.13 | **8.11** | 2.67 | 11.64 | 6.59 | 1.20 | 0.71 |
| Ours | 6.69 | 2.23 | 8.12 | **2.51** | 11.28 | 6.40 | 1.23 | 0.81 |
| FCA | 6.51 | 2.17 | 15.10 | 2.95 | 15.63 | 5.99 | 1.77 | **0.67** |
| FCED | 6.26 | 2.14 | 14.61 | 2.75 | 15.87 | **5.94** | 1.81 | 0.73 |

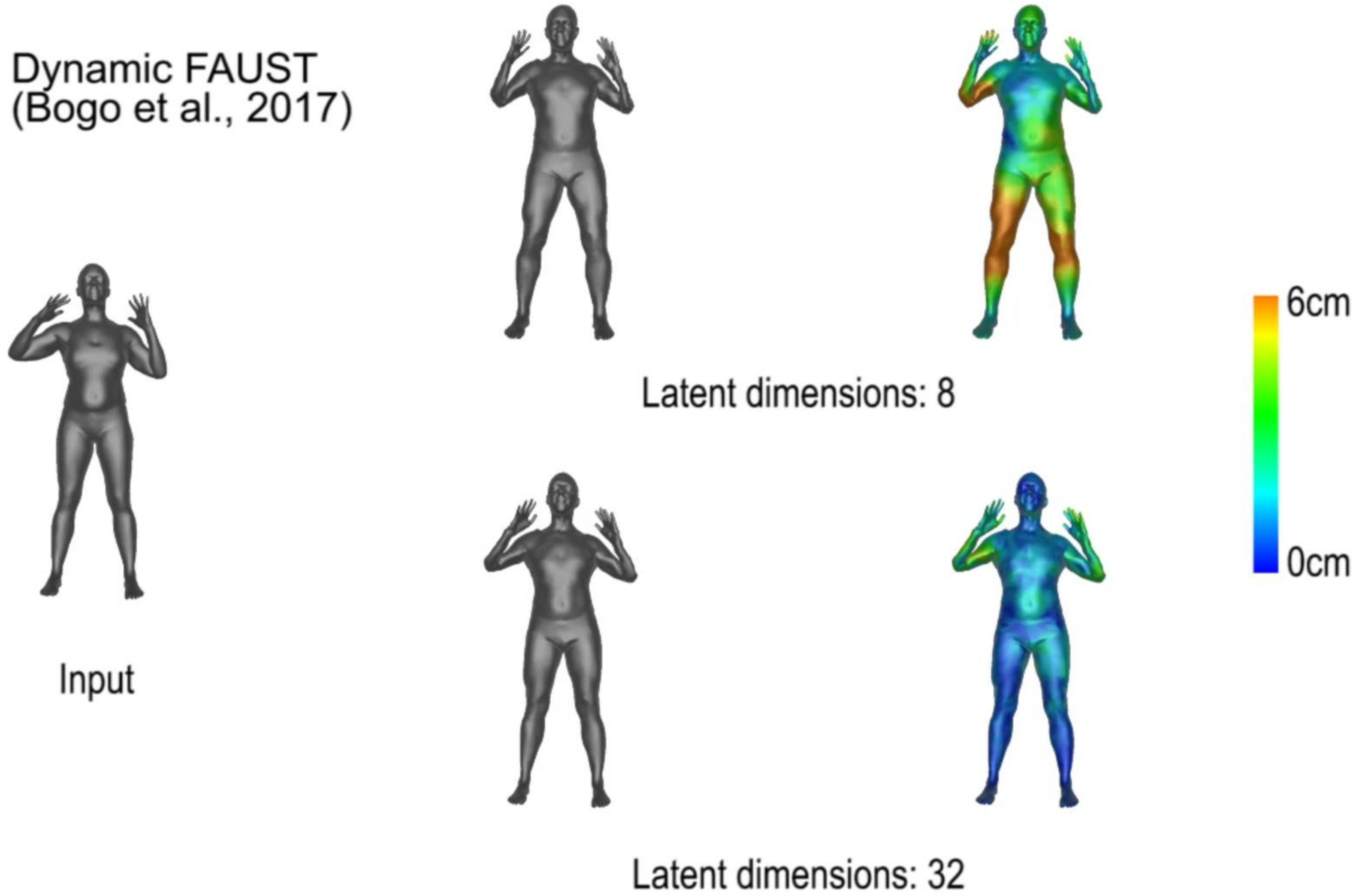
- DEMEA is quantitatively on par with convolutional baselines
  - But: unlike them, DEMEA has no artifacts
- FC baselines perform poorly on 8 latent dimensions, very well on 32

# Results: Prior Work

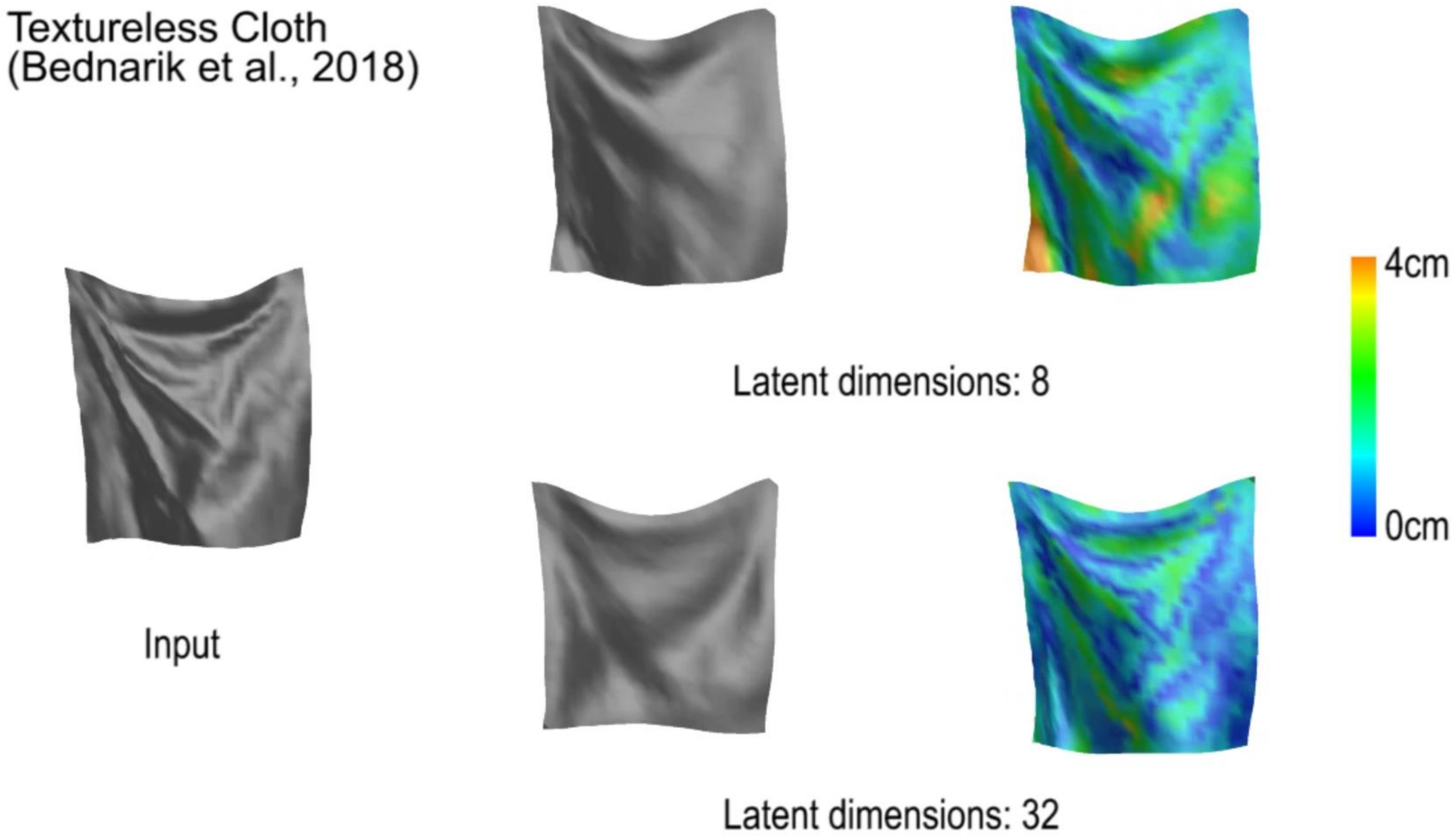
- DEMEA outperforms CoMA on all four datasets
- Neural 3DMM:

| | DFaust | | SynHand5M | | Cloth | | CoMA | |
|---|---|---|---|---|---|---|---|---|
| | 8 | 32 | 8 | 32 | 8 | 32 | 8 | 32 |
| N. 3DMM | 7.09 | **1.99** | 8.50 | 2.58 | 12.64 | 6.49 | 1.34 | **0.71** |
| Ours | **6.69** | 2.23 | **8.12** | **2.51** | **11.28** | **6.40** | **1.23** | 0.81 |

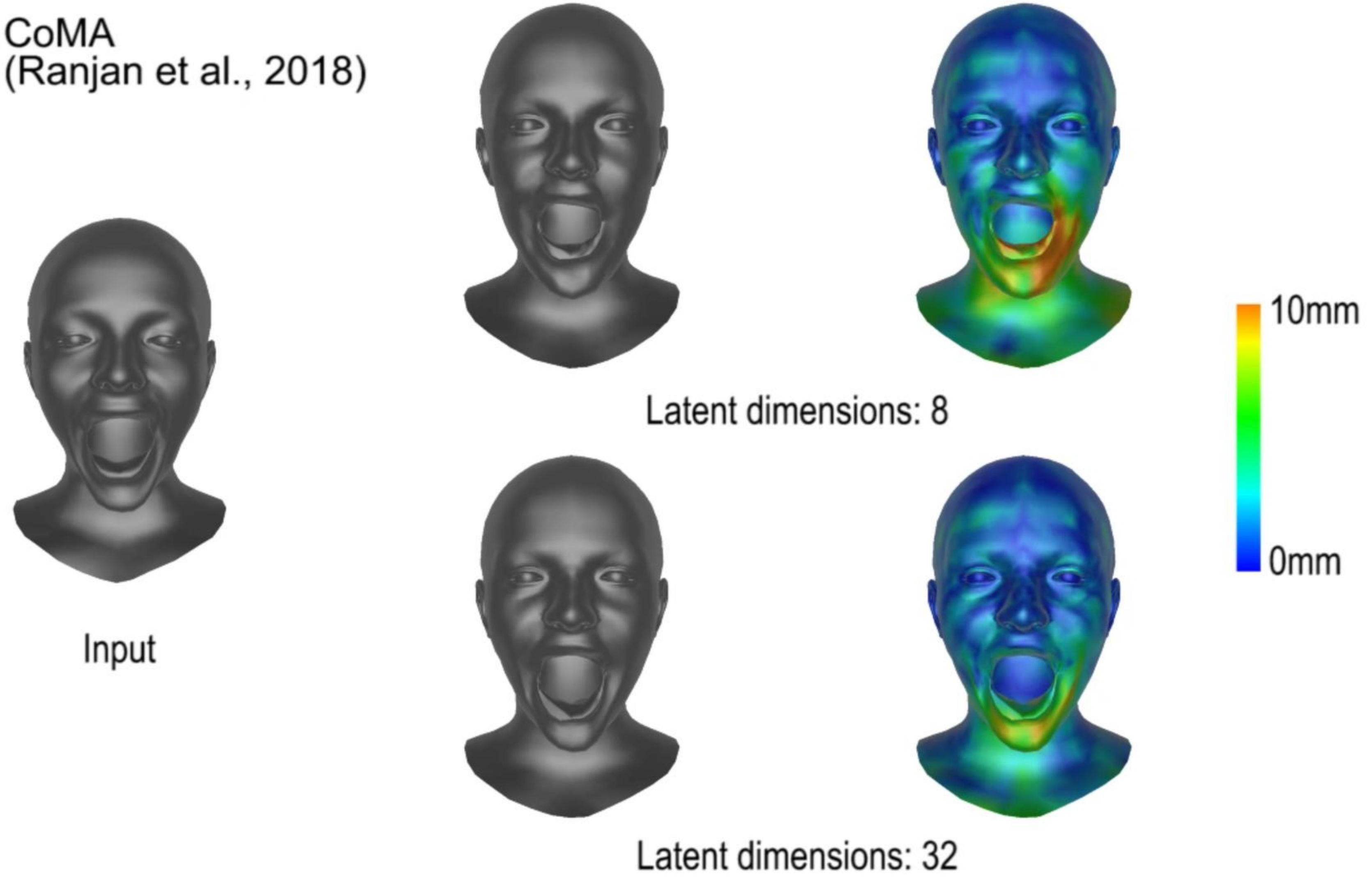
# Results: Qualitative



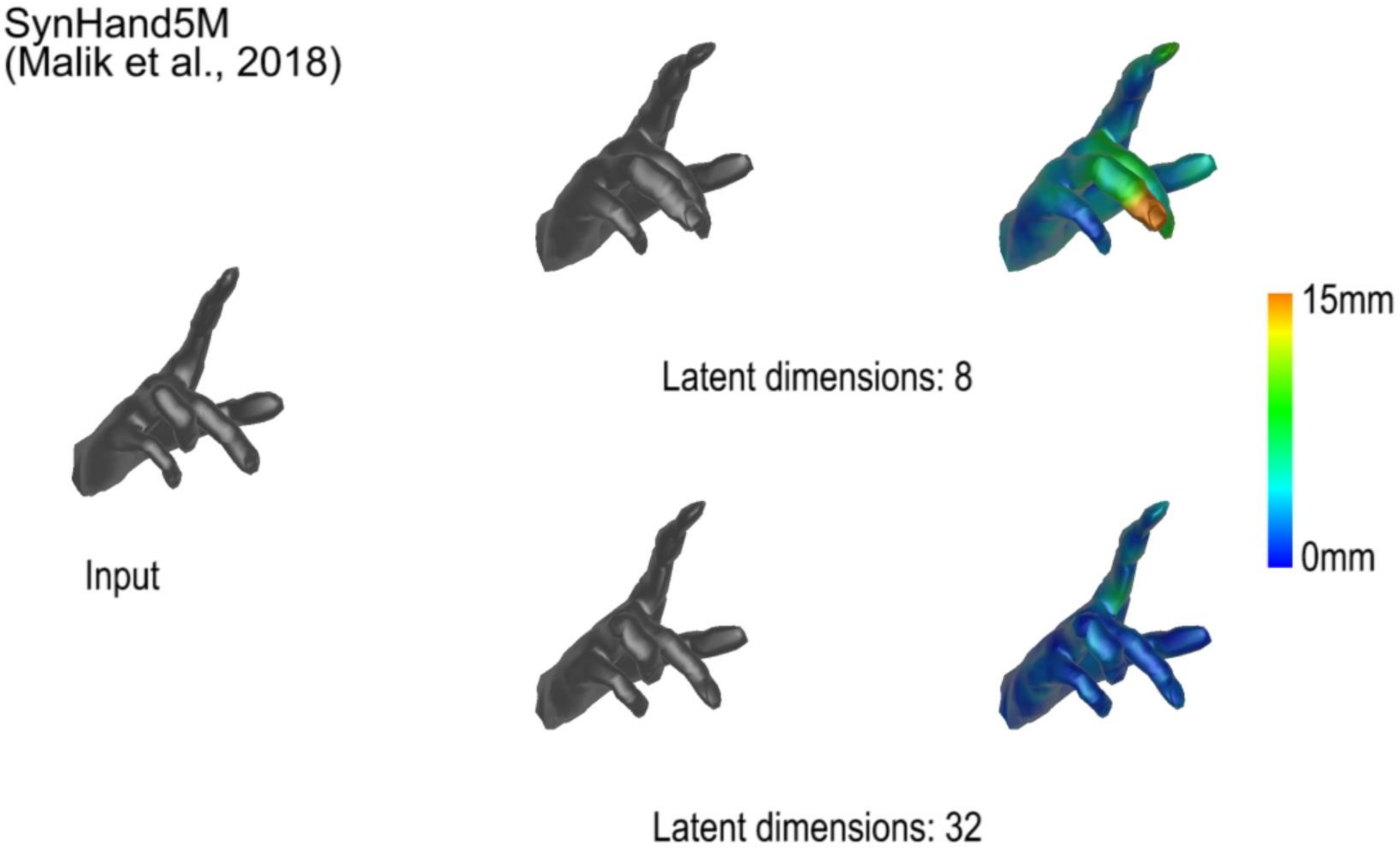
# Results: Qualitative



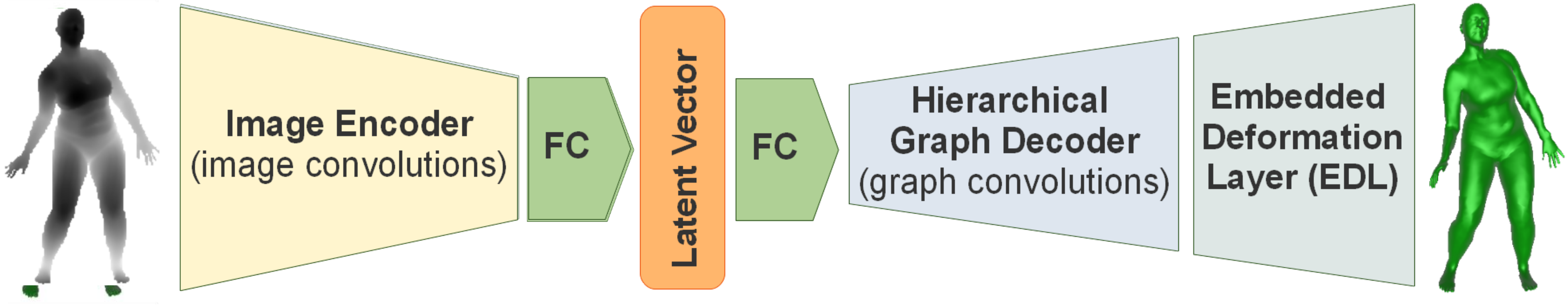
# Results: Qualitative



# Results: Qualitative

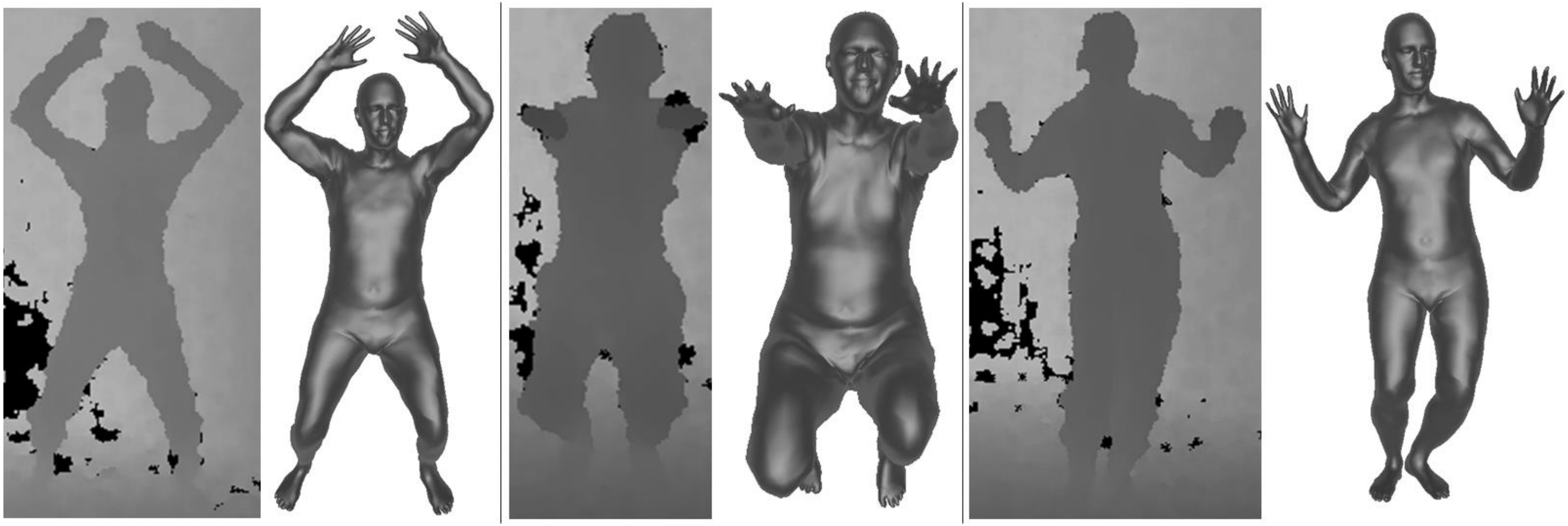


# Applications: Image-to-Mesh



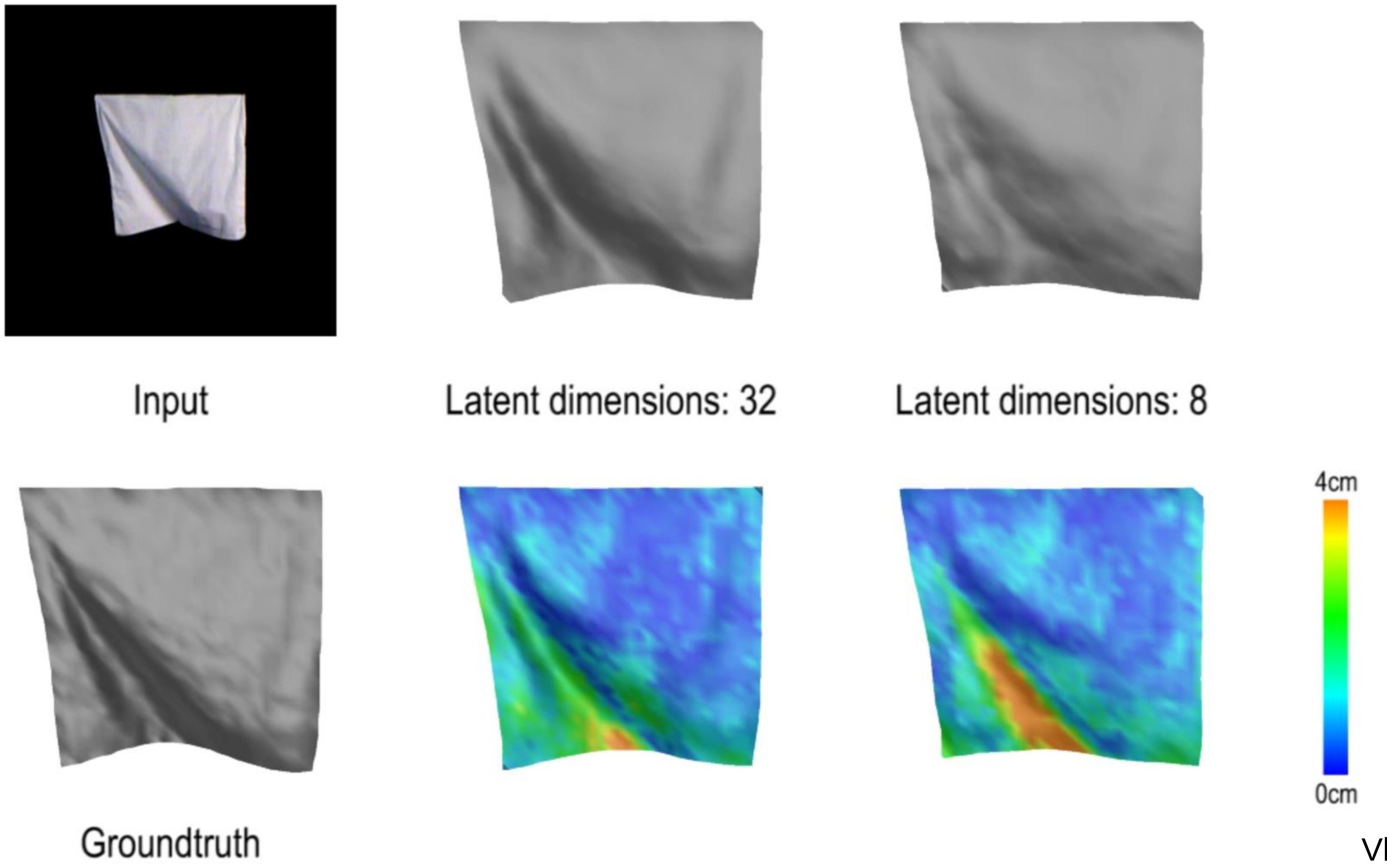
# Applications: Image-to-Mesh

- Train on augmented synthetic data
- Apply to real Kinect depth images

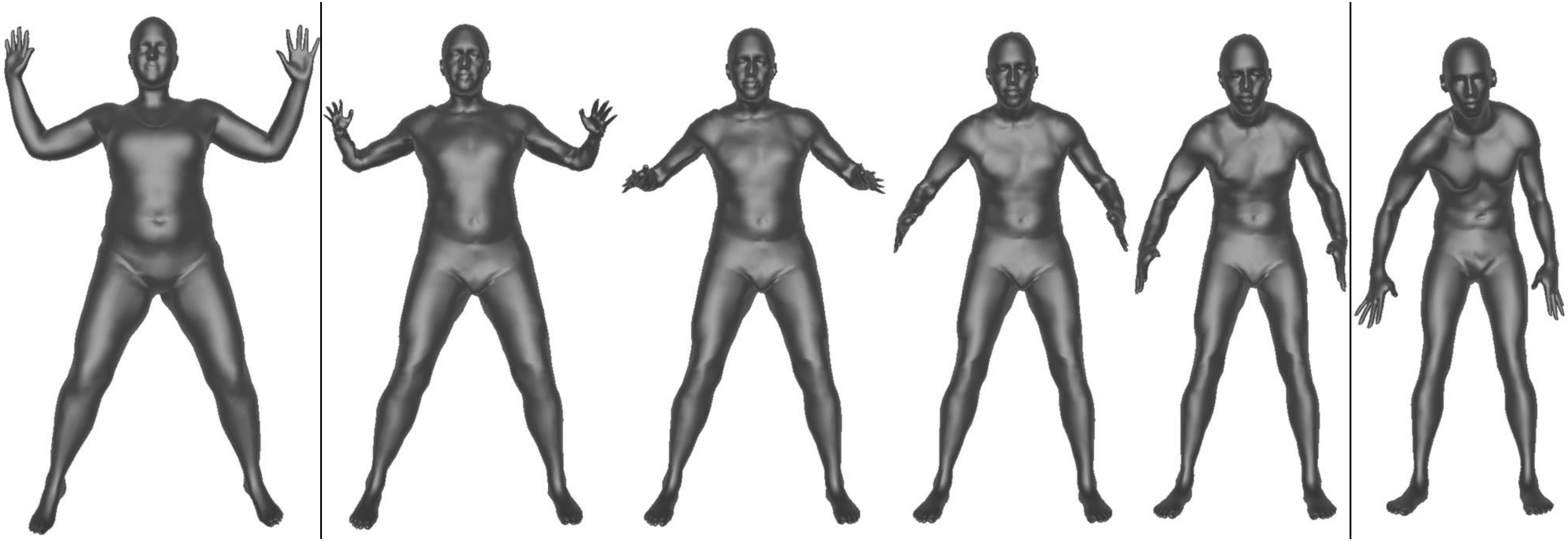


# Applications: Image-to-Mesh

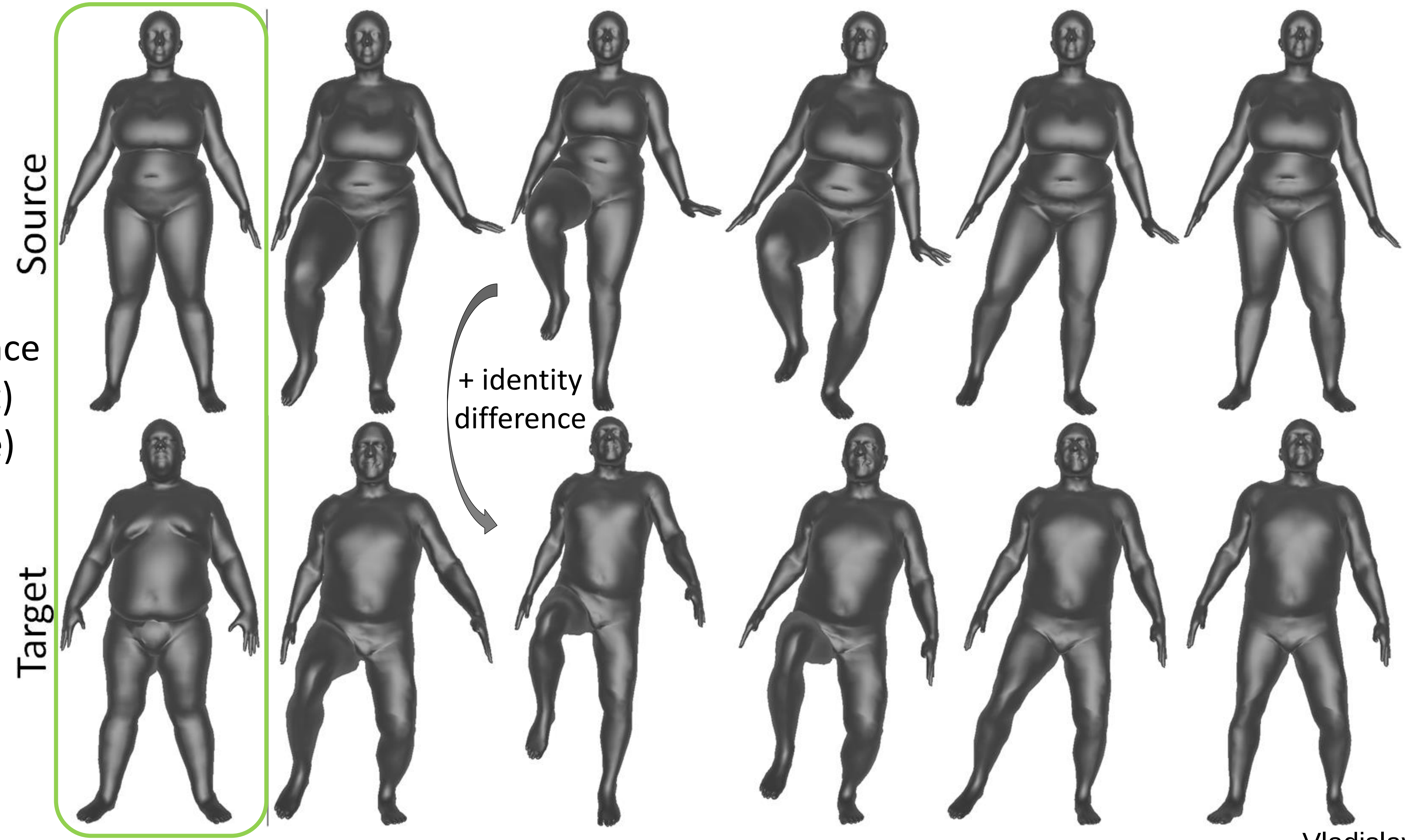
- Train and test on real RGB data



# Applications: Interpolation



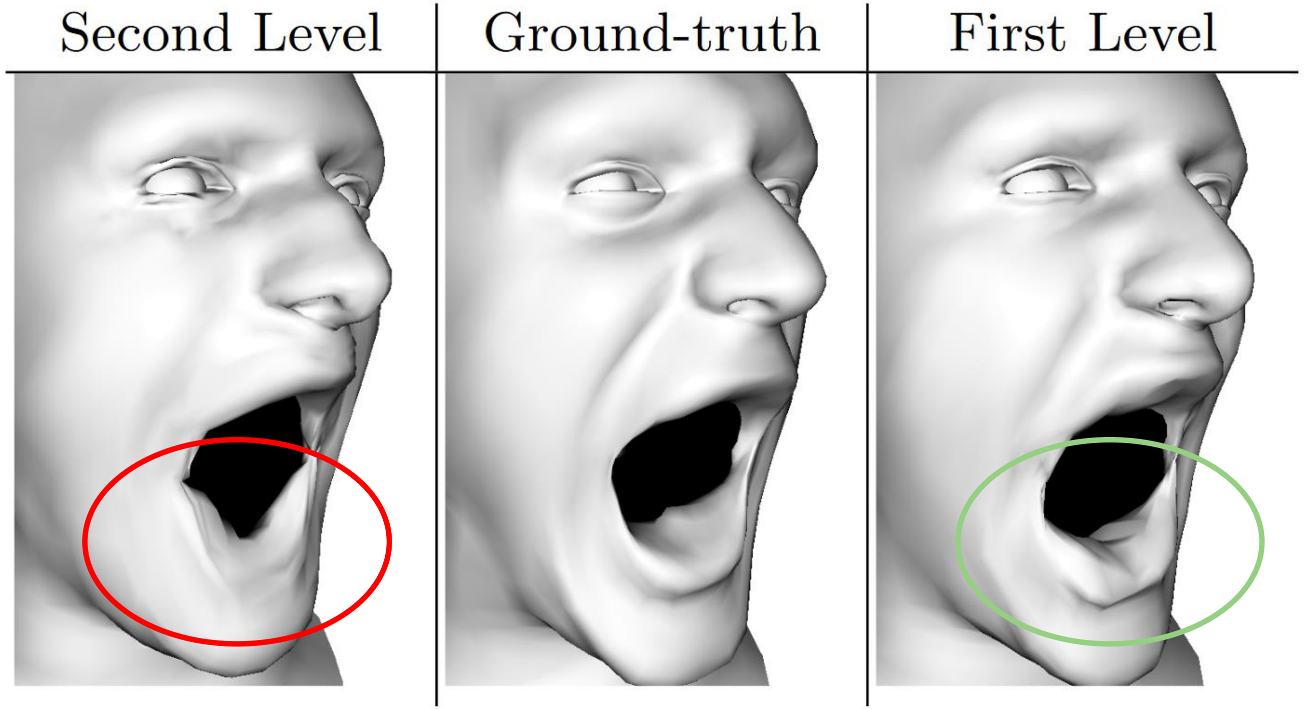
# Application: Deformation Transfer




identity difference
= latent(target)
- latent(source)

# Limitations

- Requires large datasets of *registered* meshes
- Currently trained per category
- Subtle deformations are difficult to capture
  - Similarly, embedded graphs can be too coarse



# Summary

- DEMEA achieves mesh dimensionality reduction by combining learning-based auto-encoding and explicit deformation modelling
- Advantages shown on several datasets, especially on Dynamic FAUST
- Several applications:
  - RGB-to-mesh reconstruction
  - Depth-to-mesh reconstruction
  - Interpolation
  - Deformation transfer

For more details and results, check out our paper, supplemental video and material!

# Thank you for your attention!

# References


- Bednarik et al. (2018), Learning to reconstruct texture-less deformable surfaces (3DV 2018)
- Bogo et al. (2017), Dynamic FAUST: Registering human bodies in motion (CVPR 2017)
- Bouritsas et al. (2019), Neural 3D Morphable Models: Spiral Convolutional Networks for 3D Shape Representation Learning and Generation (ICCV 2019)
- Defferrard et al. (2016), Convolutional neural networks on graphs with fast localized spectral filtering (NeurIPS 2016)
- Malik et al. (2018), Deephps: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth (3DV 2018)
- Ranjan et al. (2018), Generating 3D faces using Convolutional Mesh Autoencoders (ECCV 2018)
- Sorkine et al. (2007), As-rigid-as-possible surface modeling (SGP 2007)
- Sumner et al. (2007), Embedded deformation for shape manipulation (SIGGRAPH 2007)