UNIVERSITÀ
DEGLI STUDI
FIRENZE

Neural 3D Morphable Models:
Spiral Convolutional Networks Applied to the BU3DFE Dataset
for 3D Shape Representation Learning and Generation

Students:
Paula Mihalcea
Lorenzo Mandelli

Professor:
Alberto Del Bimbo
Advisor:
Claudio Ferrari

April 1st, 2021

Introduction

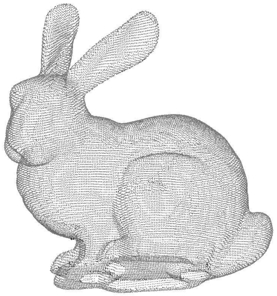
- **Automatic generation of 3D shapes** is necessary in a growing number of applications:
  - Computer graphics
  - Cinema
  - Game design
  - 3D printing…

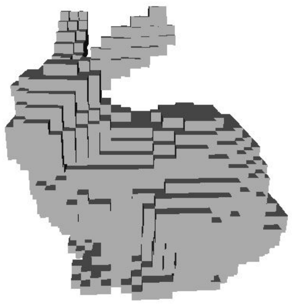
Why waste time modeling a human face from scratch when you can start from an existing model?

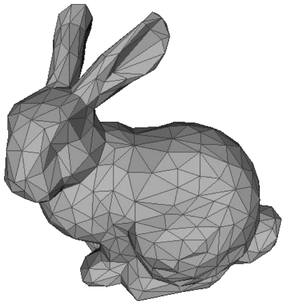
- Traditional algorithms yield satisfying results.
- AI-enabled mesh generation produces more complex, realistic and diversified 3D shapes.

The main issue

- Convolutional Neural Networks (CNNs) work well with 2D Euclidean data (e.g. images).
- There is no equivalent for non-Euclidean data structures.
- Defining a convolutional operator on 3D meshes, graphs and manifolds is hard.
- Most research focuses on alternative data representations instead of fully-triangulated meshes.



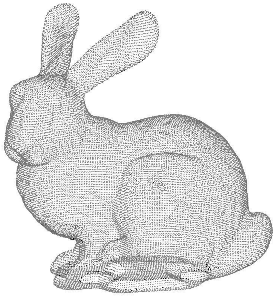
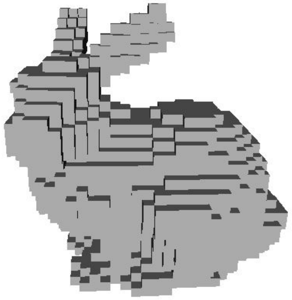
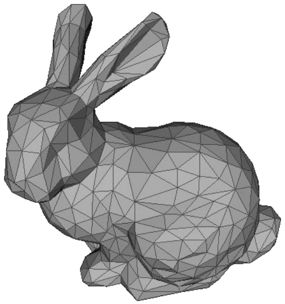




3D convolution?

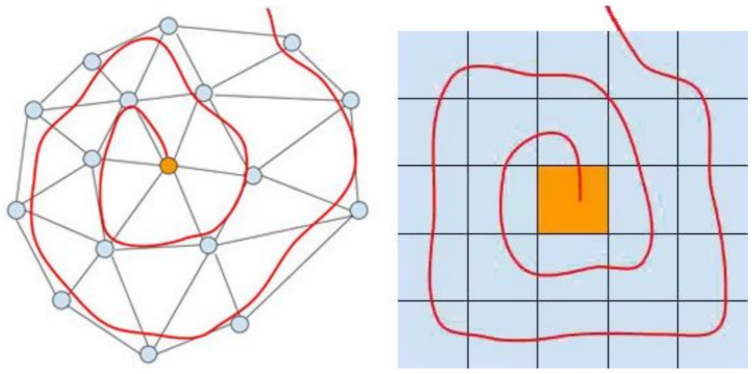
- Insensitivity to the **graph topology**
  - Solved by using fixed topology meshes in dense correspondence with a template.

- Lack of a **global ordering** of the vertices
  - Traditional methods like spectral analysis are weak on 3D meshes.

Spiral convolution

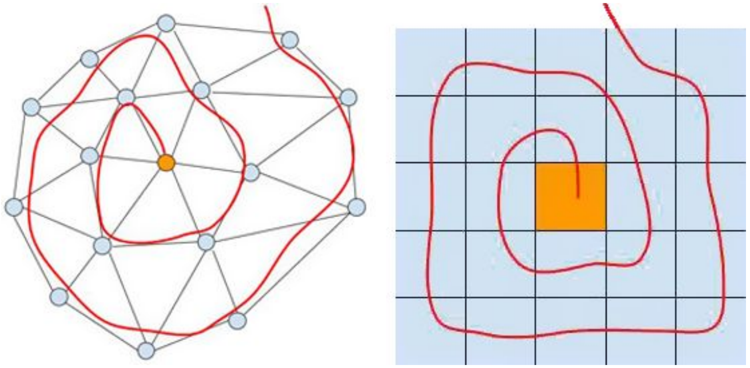
- Actually, vertices can be ordered by applying a **spiral ordering**.
- Introduced in 2019, it represents a state-of-the-art generalization of the convolutional operator to 3D shapes.
- It is an anisotropic, computationally efficient operator with notable representational power.



Spiral ordering on a mesh (left) and on a traditional 2D image (right).

Spiral ordering

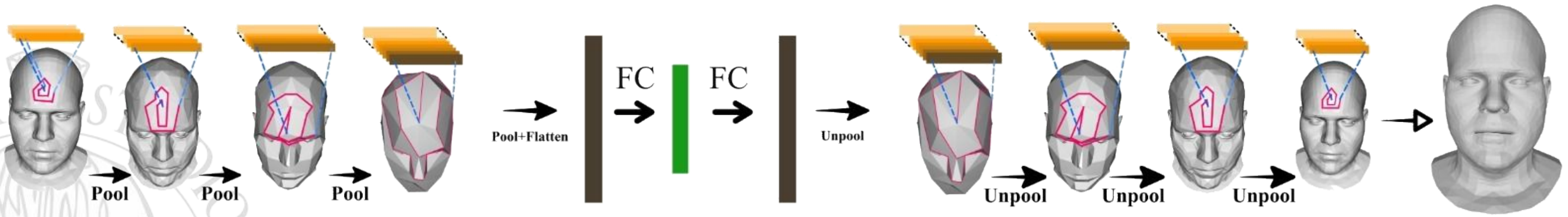
- A spiral is built for each vertex in order to process it in relation with its neighbors.
- Neighbors are ordered by fixing two degrees of freedom:
  - the **direction** (clockwise or **counterclockwise**);
  - the **first vertex**, the point in the direction of the **shortest geodesic path** to a fixed reference vertex $\mathbf{x_0}$.



Spiral ordering on a mesh (left) and on a traditional 2D image (right).

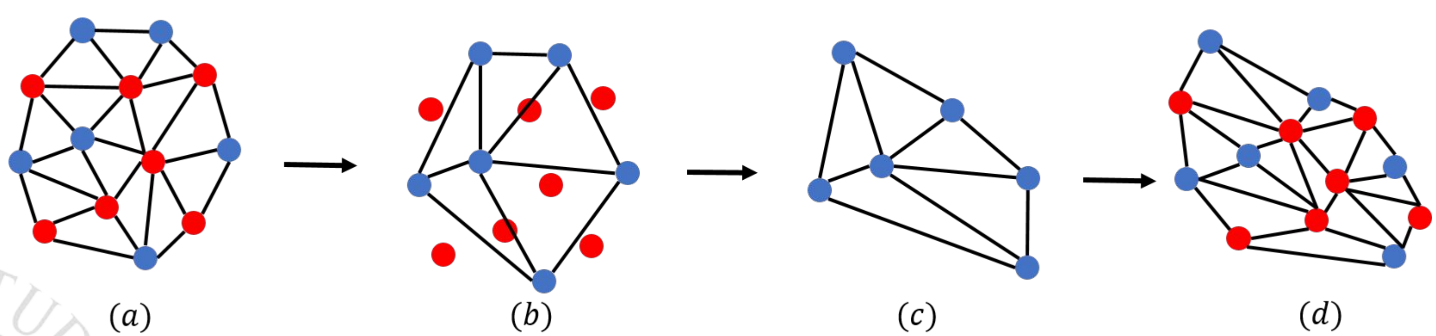
Neural 3D Morphable Model

- Neural 3D Morphable Model is a **deep convolutional mesh autoencoder**.
- It is composed by alternating **spiral convolutional** layers with **downsampling** (encoder) and **upsampling** (decoder) layers, equivalent to pooling and unpooling, respectively.
- In other words, it is analogous to a classic 2D CNN.



Mesh Sampling

- **Downsampling**: mesh decimation
  - Standard procedure in mesh processing
  - Reduces the total number of vertices
- **Upsampling**: reconstruction of the downsampled mesh
  - Actually an approximation
  - Adds new vertices obtained through interpolation
  - Neighboring vertices are weighted with barycentric coordinates



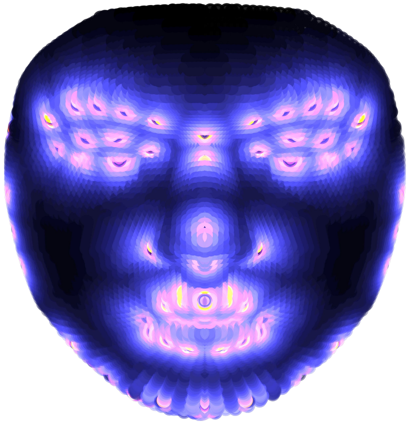
Encoder **Decoder**

Weighted L1 Loss

- We also implemented a novel **weighted L1 loss** function.
- Gives importance to the **movable regions** of the face.
- Determined by a series of **landmark points**.
- A **weight** w_i is assigned to each vertex p_i of the mesh according to its distance from the closest landmark Z_j.

$$L(\mathbf{S}^e, \mathbf{S}^g) = \frac{1}{N} \sum_{i=1}^{N} w_i \cdot \parallel p_i^e - p_i^g \parallel_1$$

$$w_i = \frac{1}{\min d(p_i, Z_j)} \forall j$$



S^e is the reference mesh, S^g its reconstruction.

Datasets

COMA

- 20,466 meshes
- 5,023 vertices
- 12 subjects, 12 expressions

Expression	No. of models	% of dataset
Bareteeth	1946	9.50
Cheeks in	1396	6.82
Eyebrow	2283	11.15
High smile	1878	9.17
Lips back	1694	8.27
Lips up	1511	7.38
Mouth down	2363	11.54
Mouth extreme	793	3.87
Mouth middle	1997	9.75
Mouth open	674	3.29
Mouth side	1778	8.68
Mouth up	2153	10.51

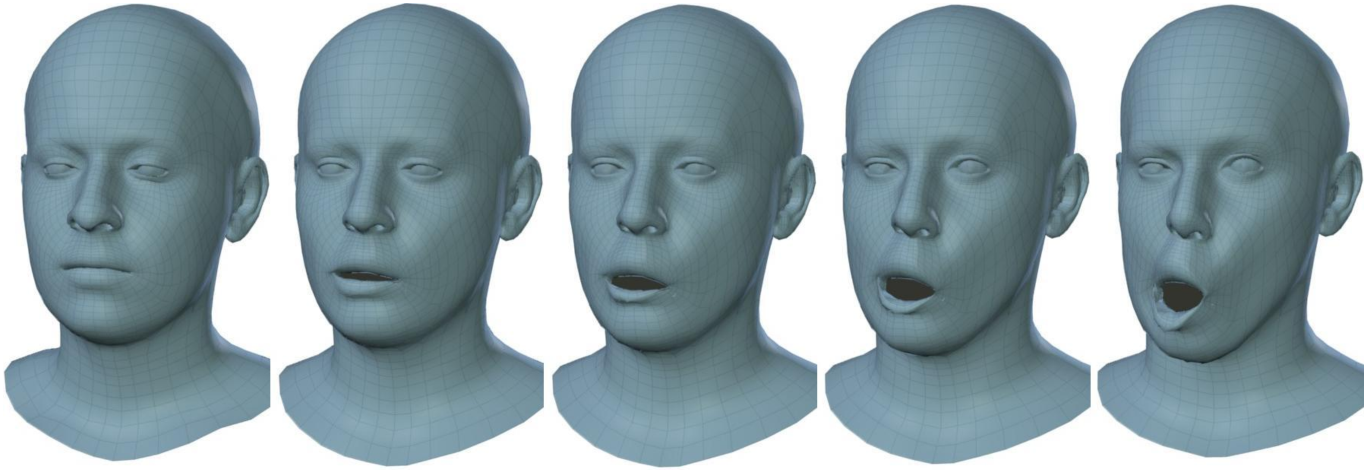
BU3DFE

- 1,779 meshes
- 6,704 vertices
- 100 subjects, 7 expressions
- Wider range of age and race

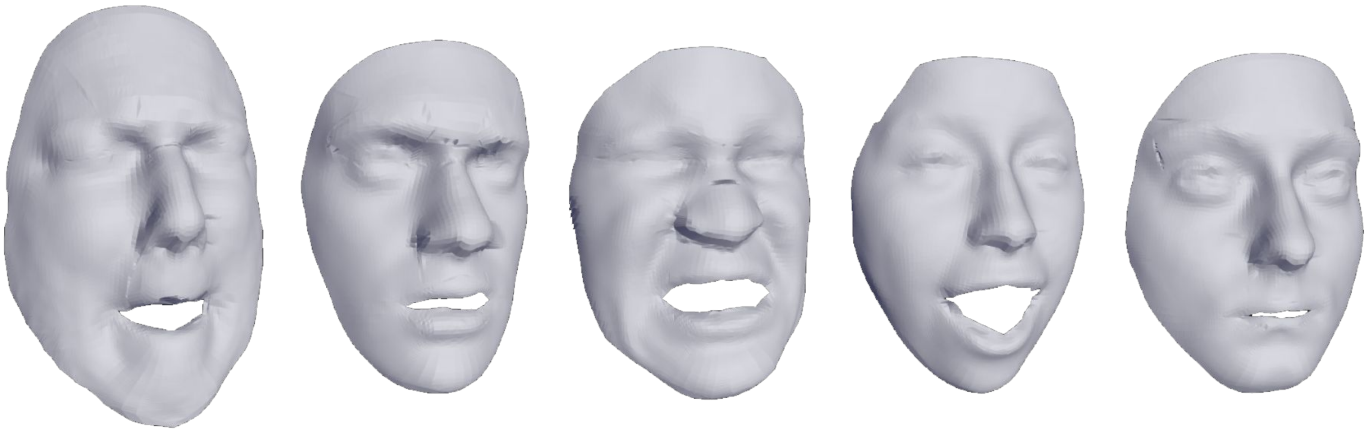
Expression	No. of models	% of dataset
Neutral	57	3.2
Angry	158	8.88
Disgust	351	19.73
Fear	349	19.62
Happy	347	19.51
Sad	161	9.05
Surprise	356	20.01

Datasets

COMA samples



BU3DFE samples



Dataset split protocols
for the BU3DFE dataset

Percentage split

- Train set: 90% of models
- Test set: 10% of models
- Lower difference between elements in the two sets
- Also used by Bouritsas *et al.*

Identity split

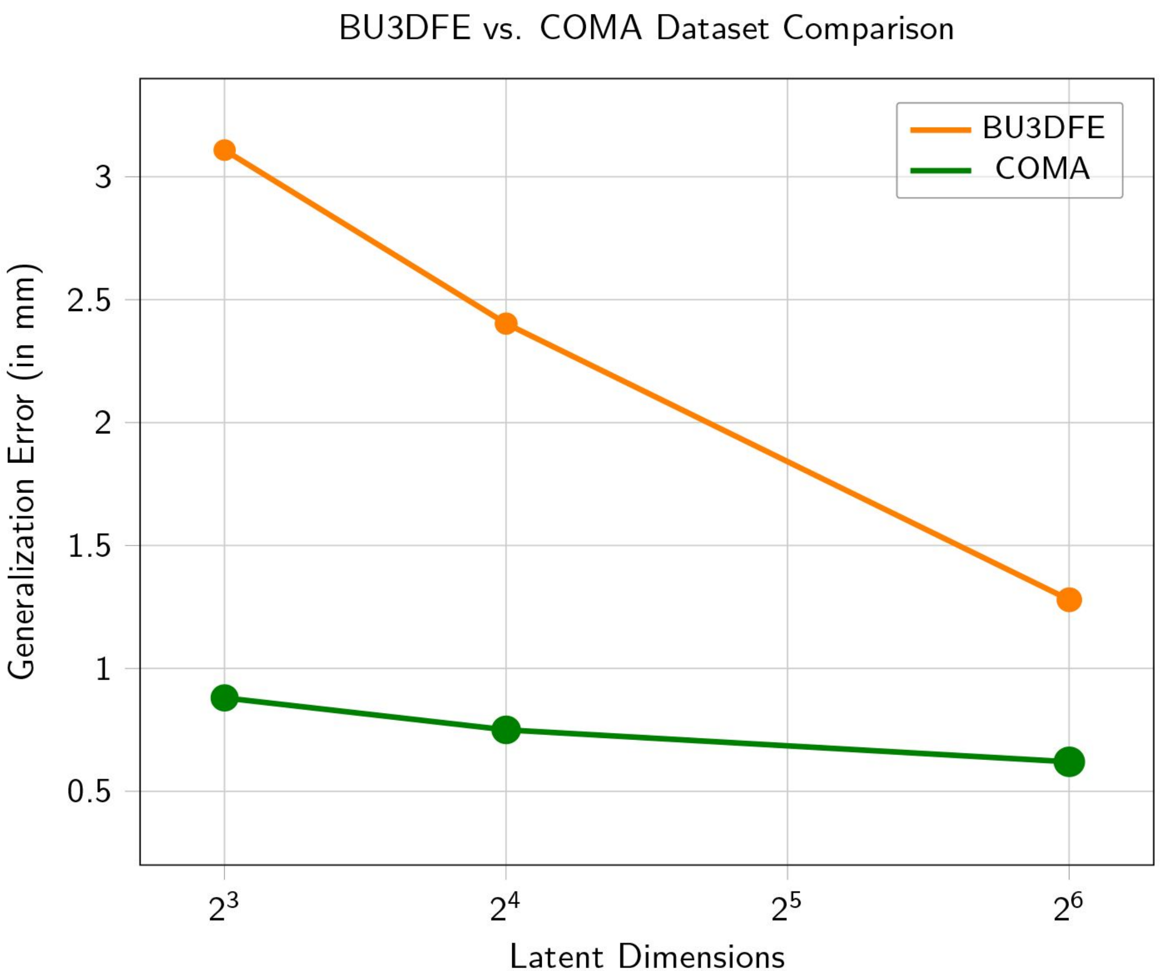
- Train set: 90% of identities
- Test set: 10% of identities
- None of the test set identities are seen by the network during training

Experiments

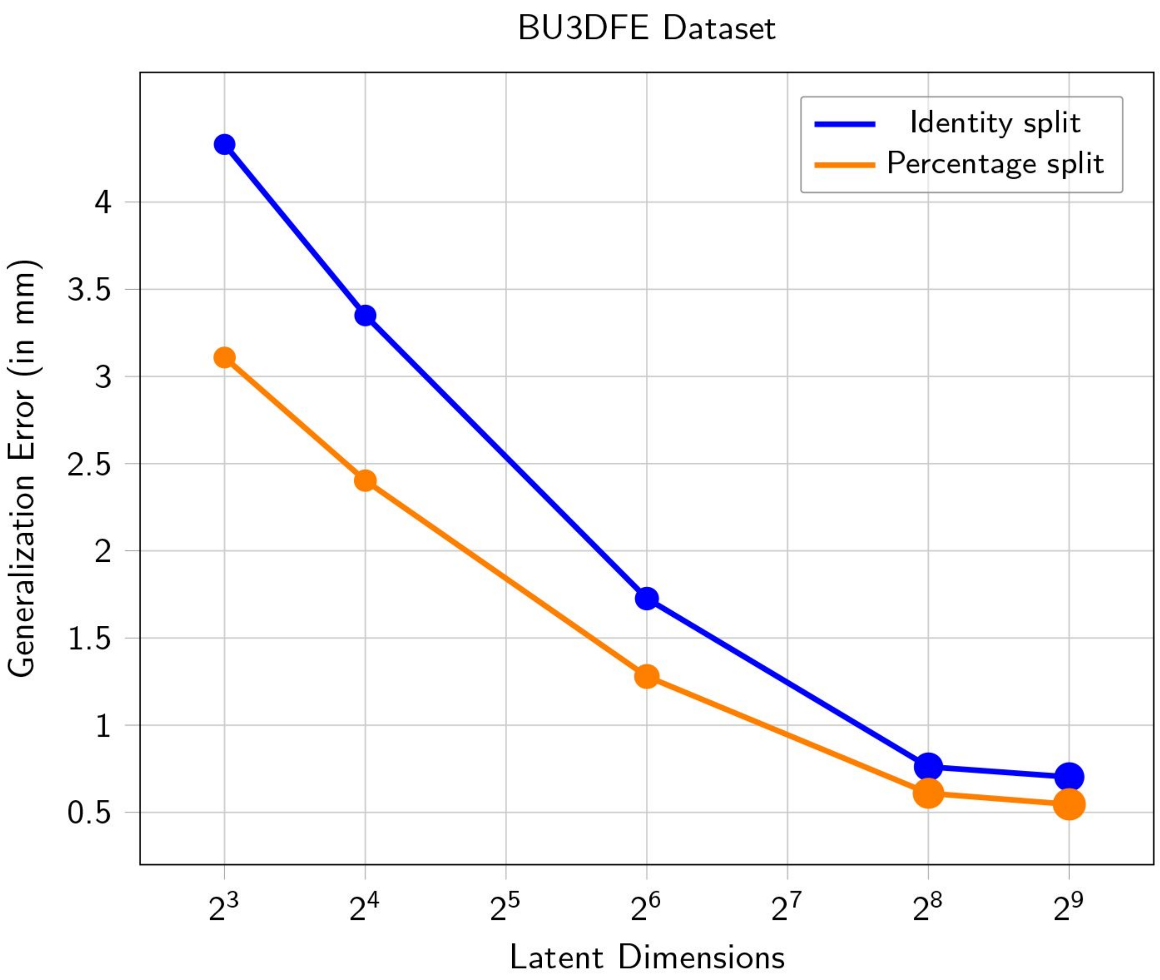
- Tested the validity of Bouritsas *et al.*'s results on the **COMA dataset**.
- Compared COMA's performance with the BU3DFE dataset.
- Explored **different dataset splits** of BU3DFE.
- Trained the network with the novel **weighted L1 loss**.

- Evaluation methods:
  - **generalisation error** (quantitative results);
  - **graphical renderings** (qualitative results).

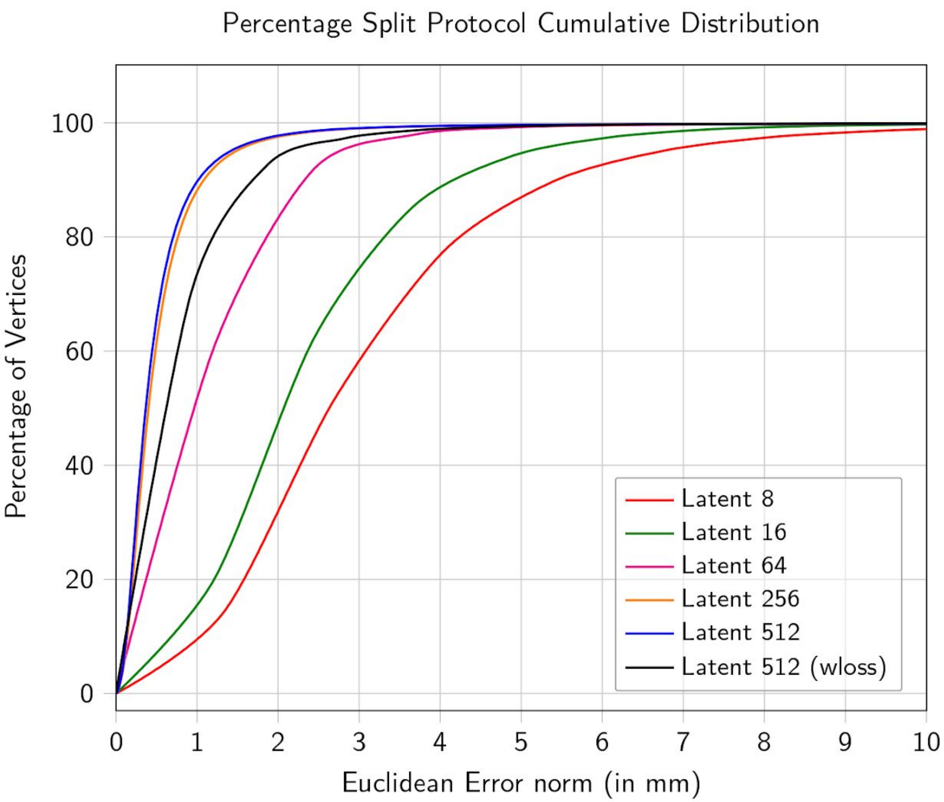
Results: Comparison to COMA

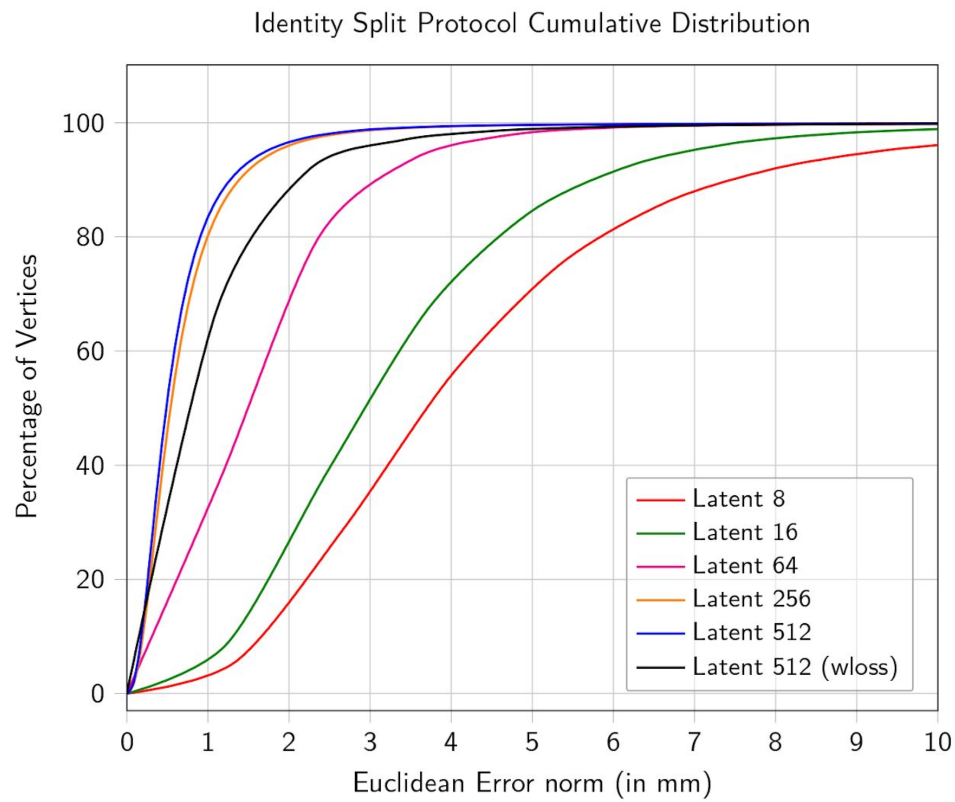


Results: Dataset split comparison

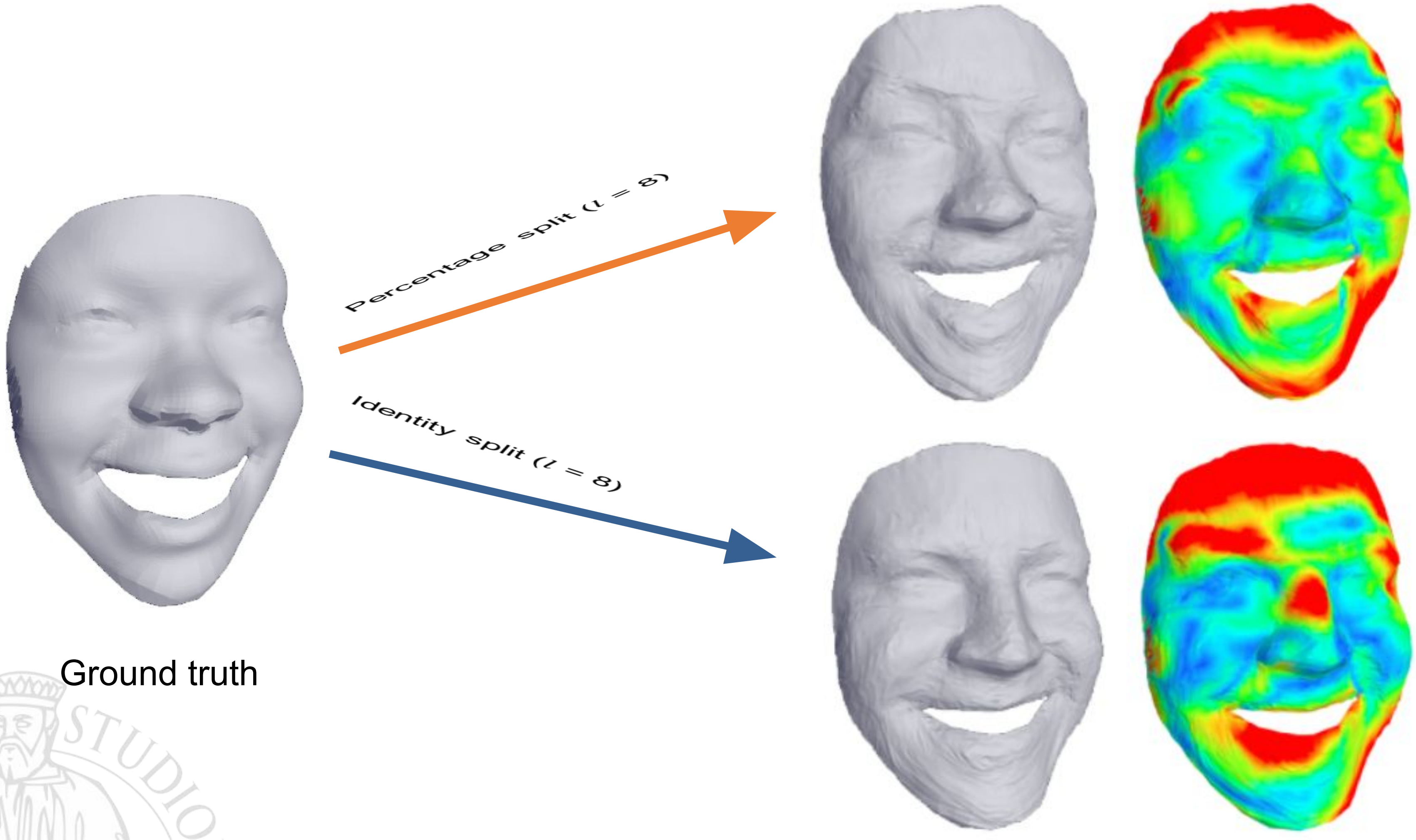


Results: Weighted loss comparison

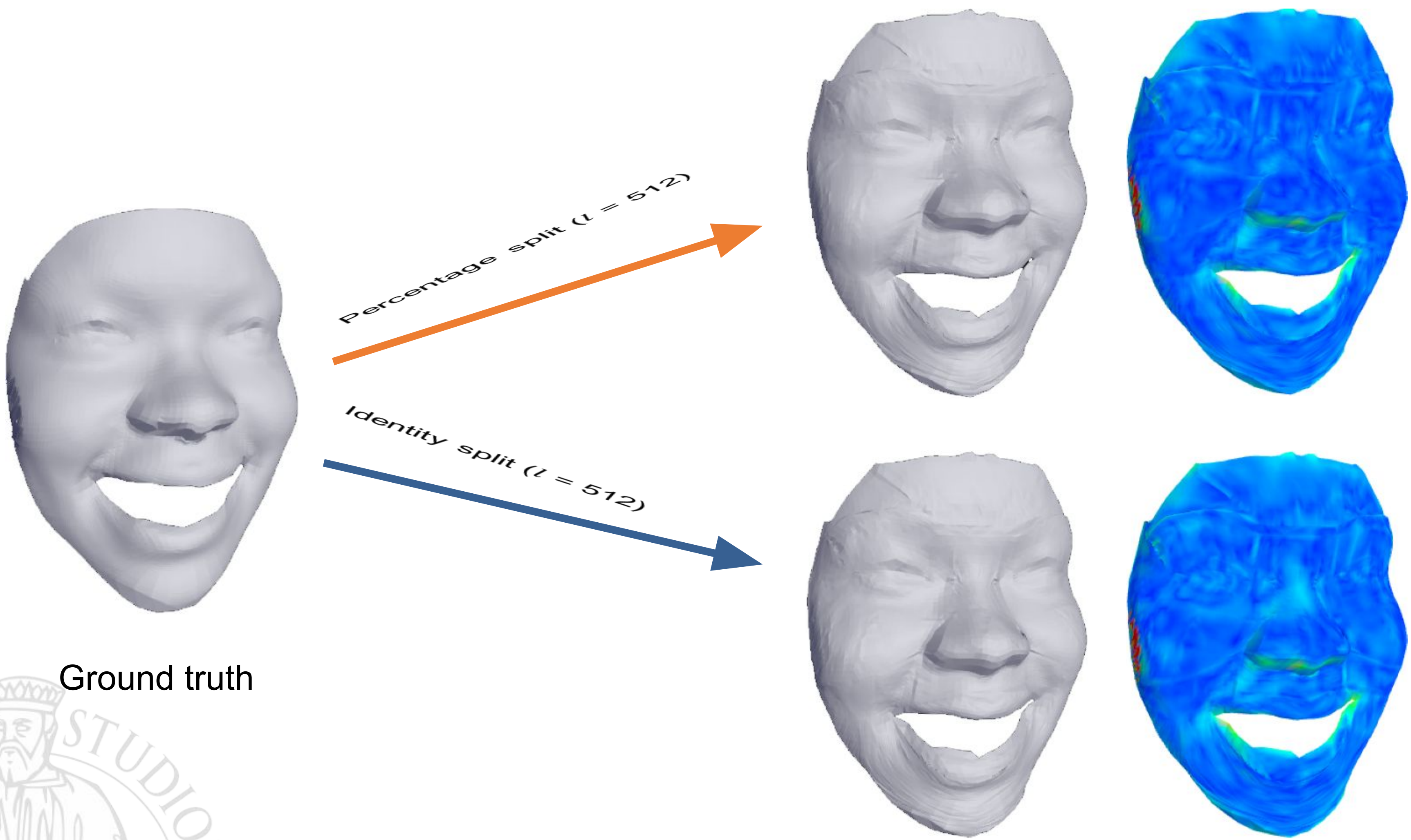




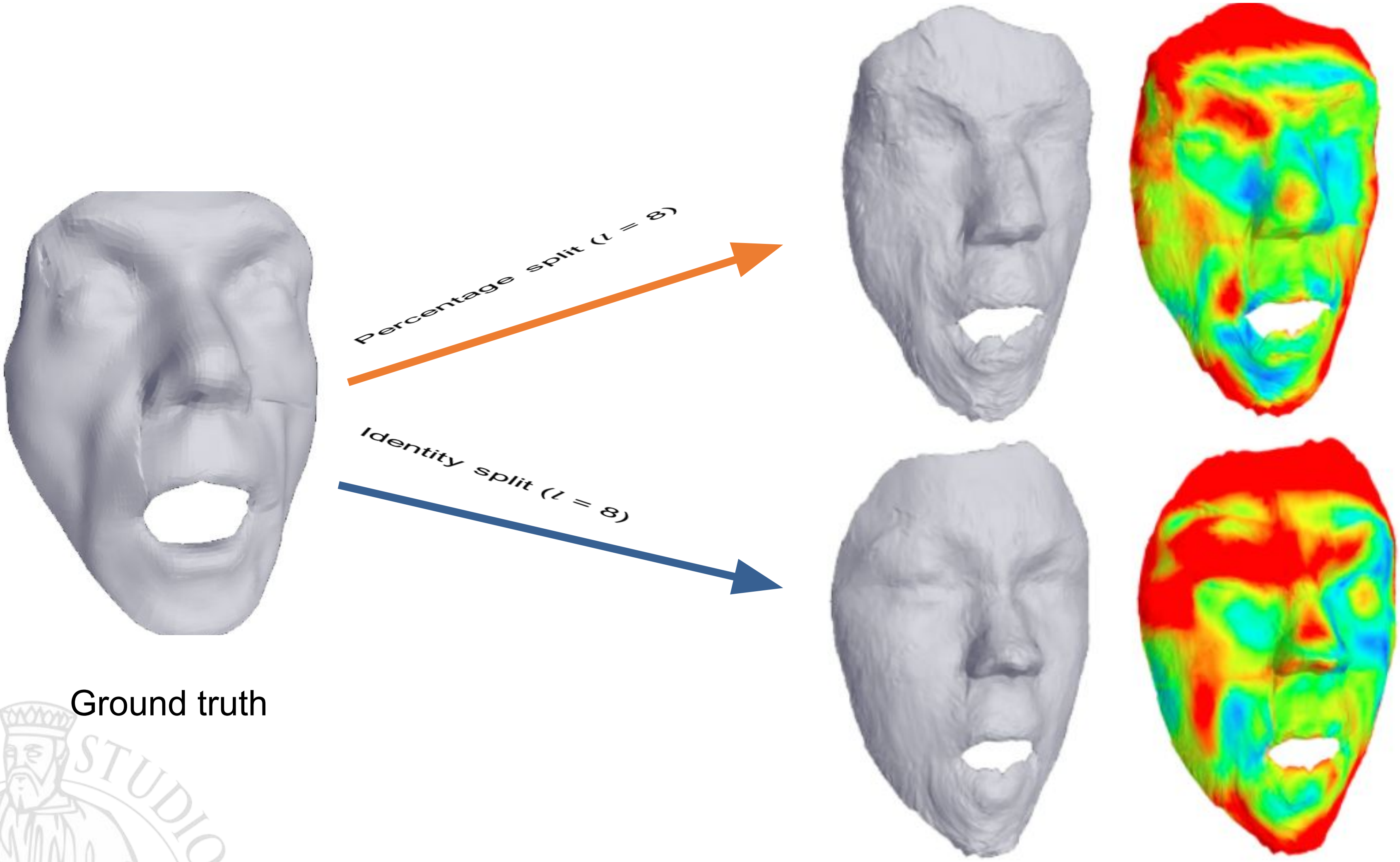
Qualitative results



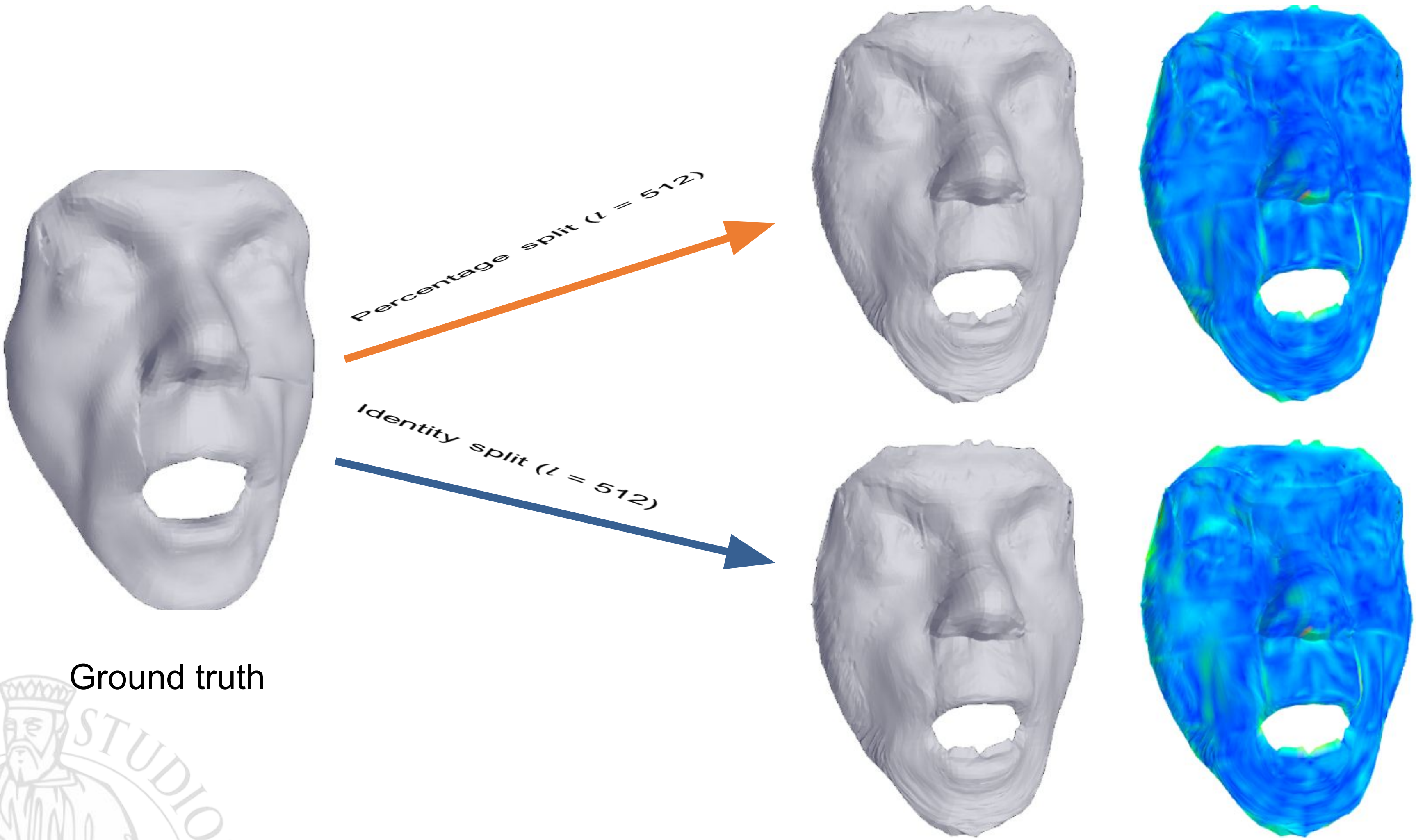
Qualitative results



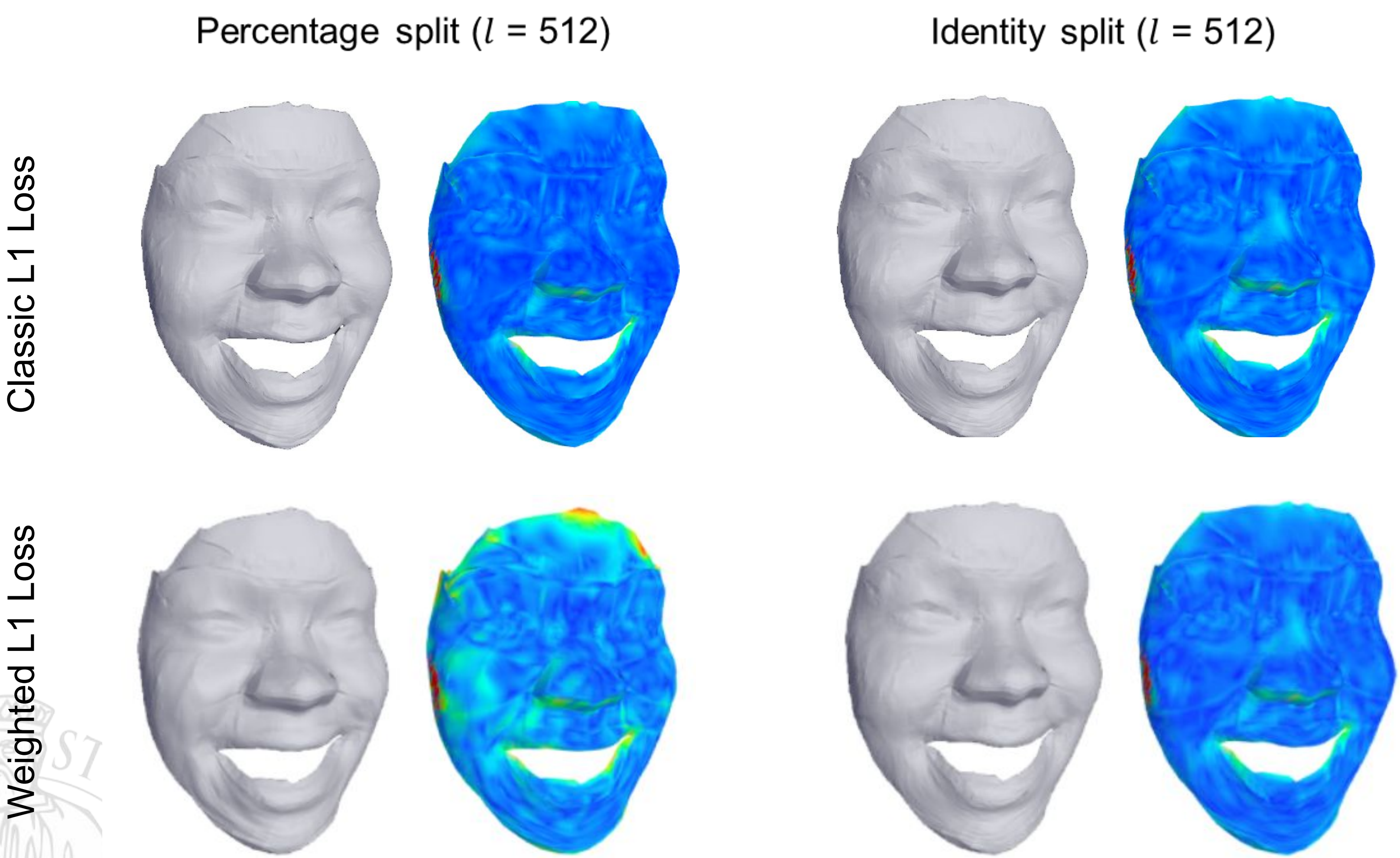
Qualitative results



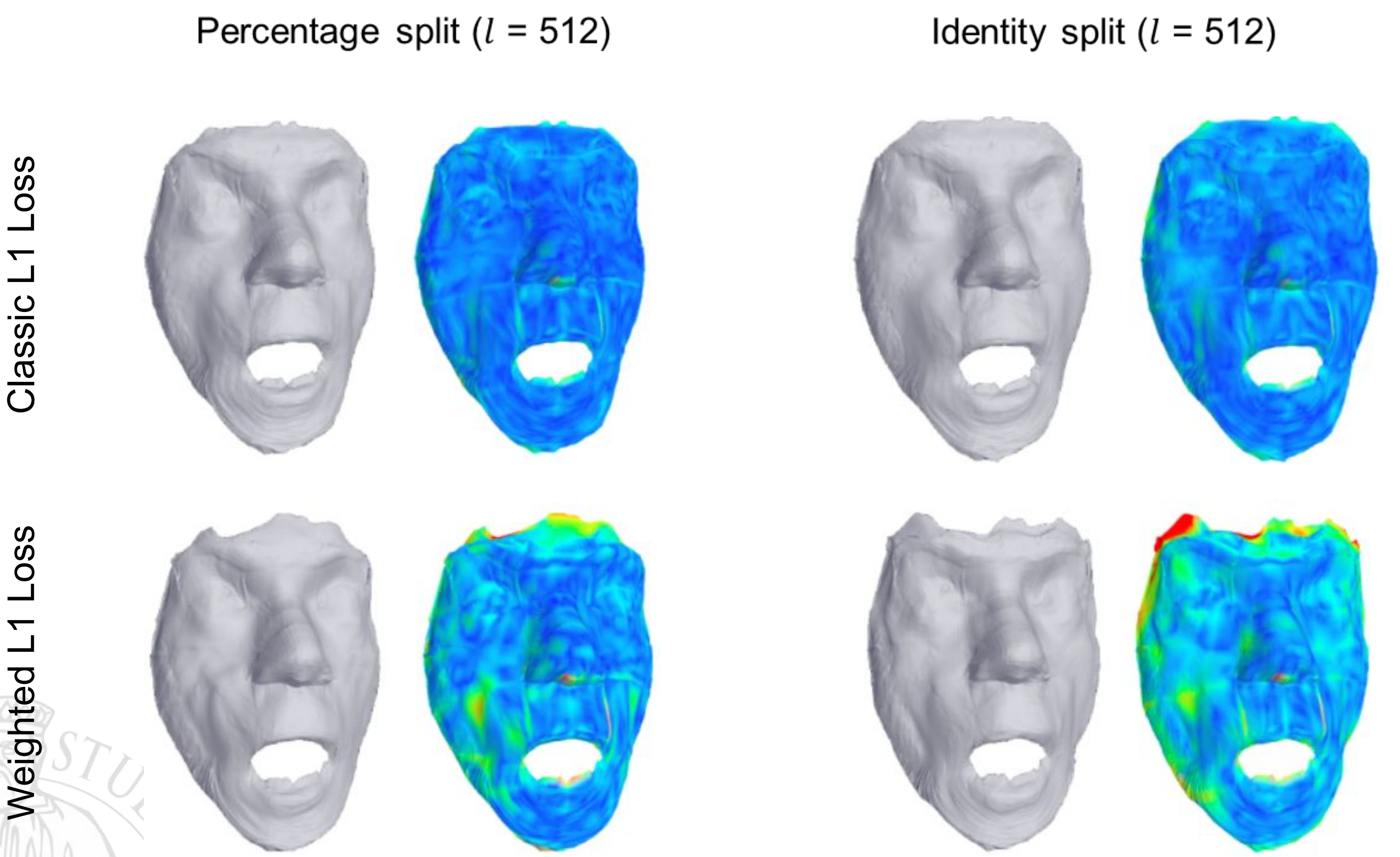
Qualitative results



Qualitative results



Qualitative results



Conclusions

- Studied **Neural3DMM**'s ability to generate 3D shapes.
- Conducted **experiments in various situations**:
  - Older, less refined data,
  - Different hyperparameters…
- **Explored** the potential of the **novel weighted L1 loss**.
  - **Landmarks focus** the training on **particular areas**.
- Showed that Neural3DMM **achieves decent results**
  - Even in non ideal conditions
  - Confirmed itself as a robust network.

Contributions and source code

- This project also led us to contribute to one of the main libraries used, namely the **Max Planck Intelligent Systems PSBody Mesh package**.
- We added a relevant extension to the library documentation.
- The MPI-IS PSBody Mesh package can now be installed on a wider range of machines.

- The complete source code of our project is freely available at https://github.com/PaulaMihalcea/Neural3DMM_BU3DFE

Thank you for your attention!