Anatomically Constrained Implicit Face Models

Supplementary Material

Figure 9. We show here the collection of 3D shapes used in our Model Learning stage. For all our experiments we used 1 neutral expression (or rest pose) and 19 expressions, all captured and reconstructed following the method of Beeler *et al*. [2].

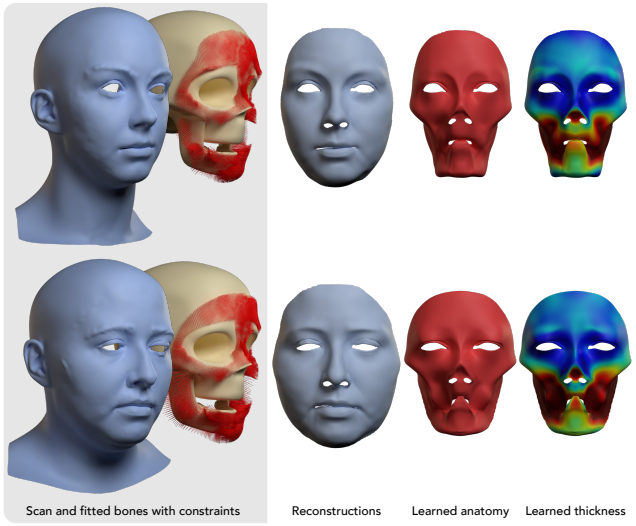

Figure 10. We show for two actors, first on the left the input neutral geometry next to the fitted skull and mandible, with an overlay of our computed sparse anatomical constraints. On the right, we show the reconstructed geometry, the learned anatomy (using those sparse anatomical constraints) and learned thicknesses.

A. Additional Details

A.1. Anatomy Constraints

We loosely regularize the skull and mandible geometries using sparse anatomical constraints. We compute these sparse constraints by fitting a template skull and mandible meshes to the neutral geometry following the method of Zoss *et al*. [56]. For any given skin point inside a hand-painted trusted region of the bone fitting process, we trace a ray along the inverse direction of the skin normal and store the bone intersection point only if the bone faces the same direction as the skin. We then trace another ray following now the bone normal, intersecting the skin again (potentially at a different point) and store the thickness and bone normal for the intersected skin point. Overall our sparse anatomical constraints exist only for 5 to 10% of the skin query points. We then use those bone points and thicknesses inside our losses $\mathbf{L}_{\mathrm{A}}$ and $\mathbf{L}_{\mathrm{D}}$ respectively. We show a visualization of the anatomical constraints and learned anatomies and thicknesses on Fig. 10. A visual depiction of the full set of 20 shapes used in our work is shown in Fig. 9.

A.2. Network Architecture

In Fig. 11 and Fig. 12, we show a detailed breakdown of our memorization and fitting networks.

B. Additional Results

B.1. Face Reconstruction from 2D Landmarks

In the main paper, we describe how to formulate a 2D position constraint to fit our anatomical implicit face model to landmarks obtained from a pre-trained landmark detector. In Fig. 13, we show qualitative results of fitting our trained anatomical implicit model to 10,000 dense landmarks predicted by a 2D landmark detector [13] on an input monocular video.

B.2. Learning Actor Specific Anatomical Properties

In Fig. 15, we show additional results of the recovered dense anatomical properties on a number of actors with varying face shapes spanning different ethnicities, and age groups.

B.3. Runtime Analysis

Our model fitting stage, which involves the training of the fitting MLPs F_W and F_T (see the main text), takes at-

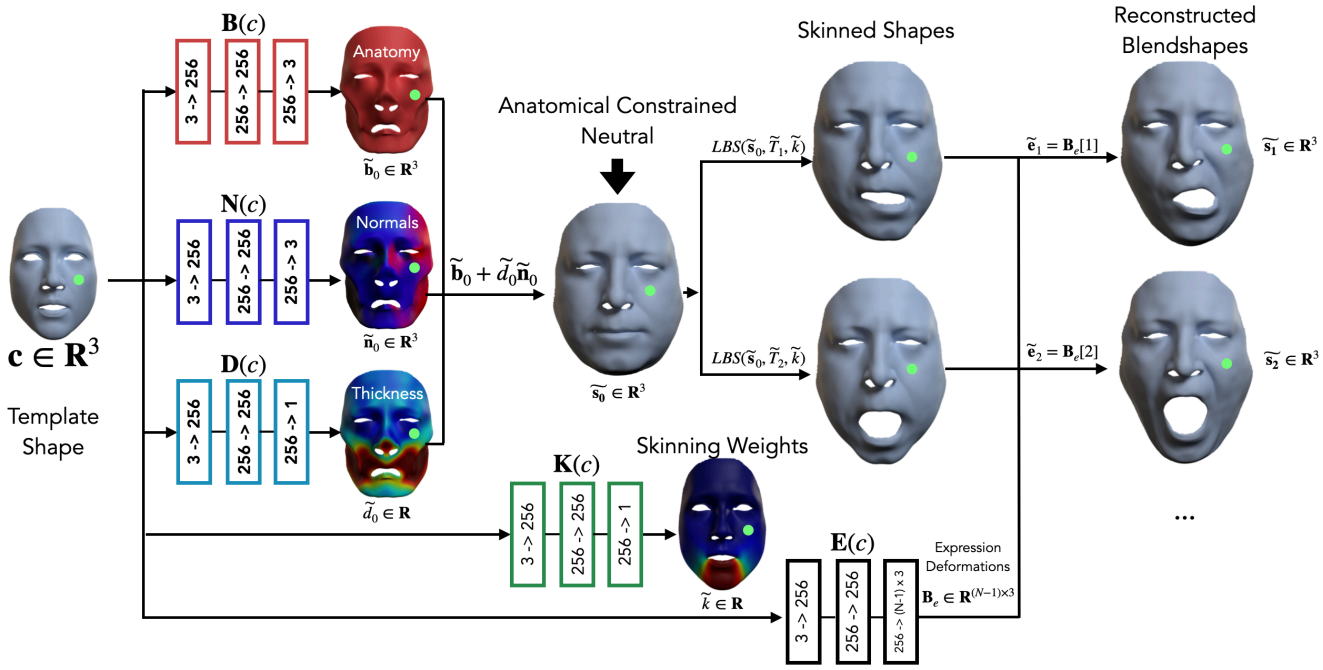

Figure 11. Starting from a query point $\mathbf{c}$ on the template shape, an ensemble of Siren MLPs [40] predict the dense underlying anatomy $\widetilde{\mathbf{b}}_0$, anatomy normals $\widetilde{\mathbf{n}}_0$, and the soft tissue thickness $\widetilde{d}_0$, using which a neutral shape $\widetilde{\mathbf{s}}_0$ of the actor is reconstructed. Then using learned per-shape jaw transformations $\widetilde{T}_i$, and actor specific skinning weights $\widetilde{k}$, the neutral is skinned to account for the rigid jaw movement. Finally, expression specific deformations $\widetilde{\mathbf{e}}_i$ are added on top of the skinned mesh to reconstruct the given blendshapes.

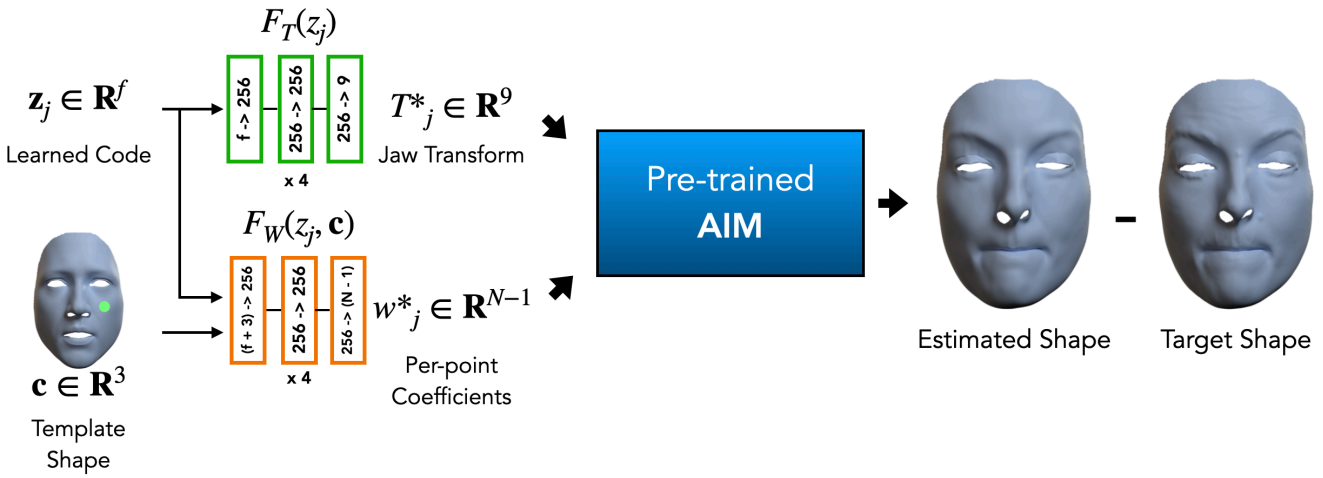

Figure 12. Given a query point $\mathbf{c}$ and a learned code $\mathbf{z}_j$ for each target shape, we use small fitting MLPs to predicts the jaw transformation $\widetilde{T}_j^*$ and the per-point coefficients $\widetilde{\mathbf{w}}_j^*$, using which the AIM model can be evaluated to result in the estimated shape. The fitting MLPs are trained to minimize the reconstruction error between the estimated and target shape.

most a few seconds per-frame to converge on a Nvidia RTX 3090. As an engineering update to our system, we experimented with the *tinycuda* framework of Muller *et al*. [32] and found that it provided a 2x performance improvement in model fitting, without any adverse effects on fitting accuracy. We leave a more thorough performance optimization of our pipeline to future work, which could also include exploring fused MLPs for the model learning stage.

B.4. 3D Performance Retargeting

We kindly refer you to our supplemental video for additional retargeting results and qualitative comparisons.

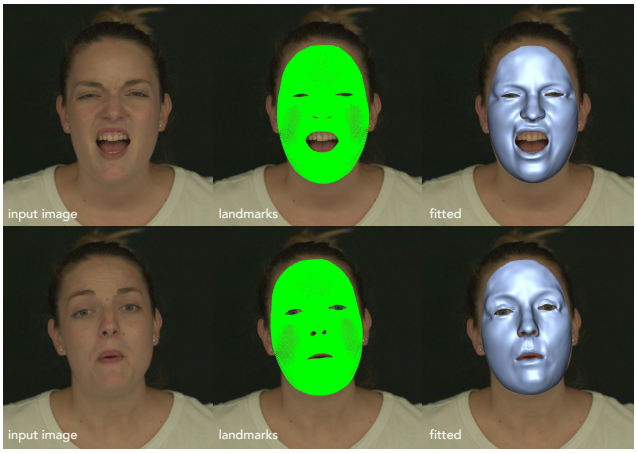

Figure 13. We demonstrate a proof of concept of the application of our model in face reconstruction, where our AIM model can be fit to 2D landmarks obtained from a pre-trained landmark detector, capturing both the pose and expression of the person faithfully.

Table 2. Average error (in mm) on a sequence of 100 frames using different types of activation functions in our MLPs.

gelu	**relu**	**siren**
0.71	0.62	0.21

Table 3. Average error (in mm) on a sequence of 80 frames using variation of our loss functions during the model learning stage.

no L_A	**no L_K**	**no L_{Sym}**	**no L_D**	**L_{Model} (Ours)**
0.29	0.22	0.24	0.21	0.19

B.5. Ablations

In Table 2 we show an ablation study on our choice of activation in our MLPs during the model learning stage. We compute the average fitting error with 3 variant of our model (each with a different activation function used in all MLPs). The **siren** variant was selected as it performs the best. Table 3 shows an ablation of the various losses used during our model learning stage. We compute the average error during the fitting stage using AIM models trained by leaving out different loss functions at the model learning stage. The fitting error is computed on a sequence of 80 unseen frames by keeping the other fitting parameters identical. We provide visual results for the several ablations we performed in our work, which include the effect of removing certain regularizers used during the model learning stage (see section in the main text) in Fig. 14, the effect of different activation functions in Fig. 16, and the size of the hidden layers used during model learning in Fig. 18.

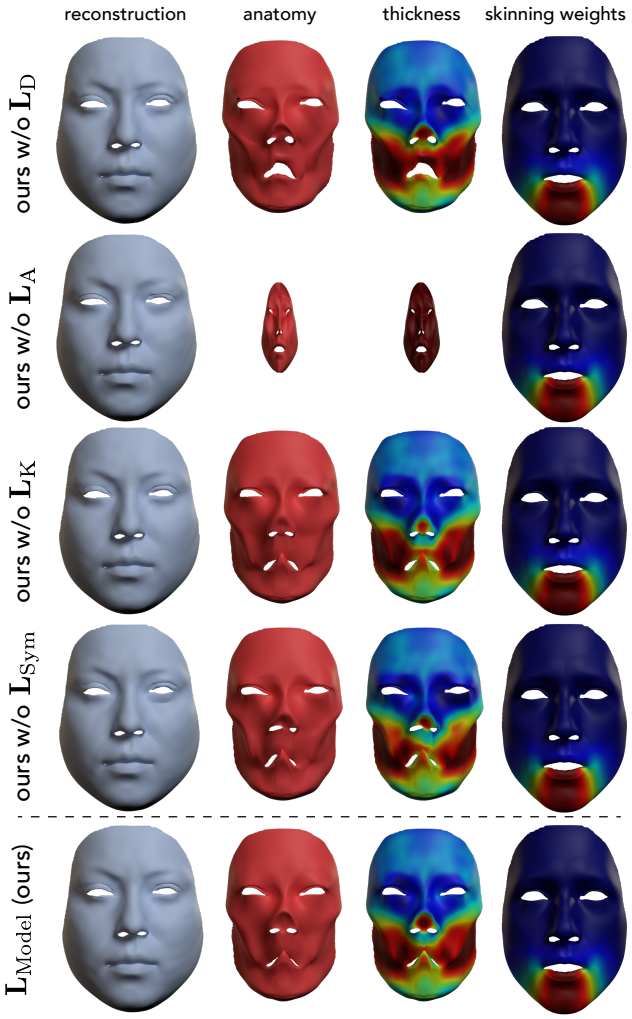

Figure 14. 1st row: We show the effect of removing the thickness regularizer L_D that encourages the soft tissue thickness to remain small in unconstrained areas, 2nd row: the effect of removing the anatomy loss L_A which result in a collapse of the learned anatomy, while still reconstructing the neutral in the first column, 3rd row: The effect of removing the optional skinning weight regularizer L_K which does not adversely affect the learned skinning weights as seen in the last column, 4th row: The effect of removing the symmetry regularizer on the anatomy, as a result of which the anatomy no longer remains symmetric, and last row, our L_{Model} loss which uses a weight sum of all regularizers.

B.6. Generic Model Comparison

As discussed in the main text, a quantitative comparison of our actor specific model against a generic 3D morphable model would be unfair to general 3DMMs as they serve a more diverse purpose. However in Fig. 17 we show a visual comparison of 2 expressions fitted using 3D positions as constraints with our model and the FLAME model [29] for 2 different

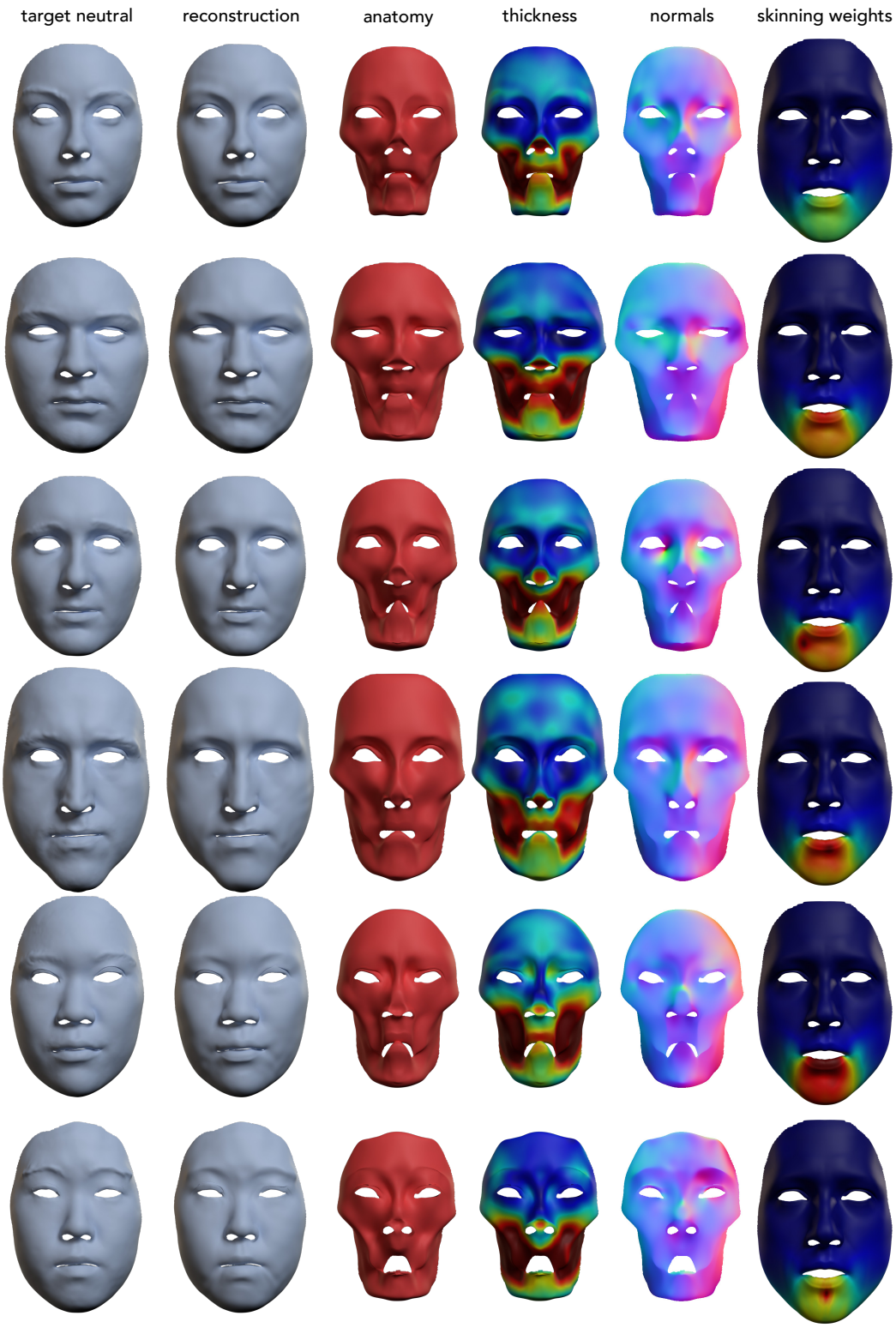

Figure 15. We show the anatomical features recovered by our formulation across a wide variety of actors. From left to right, we show the ground truth neutral shape, the reconstructed neutral shape, our learned anatomy, our learned soft-tissue thickness, our learned anatomical normals, and our learned subject specific skinning weights.

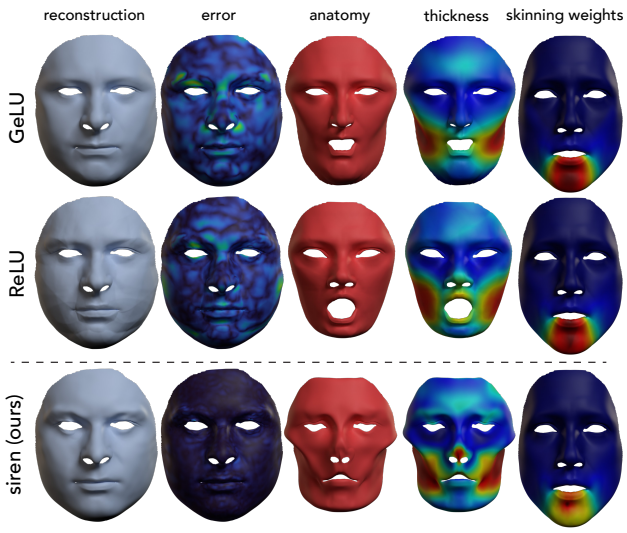

Figure 16. Using GeLU and ReLU activations in our implicit MLPs results in oversmoothed anatomy and reconstructions lacking surface detail. Sine activations provided the best results.

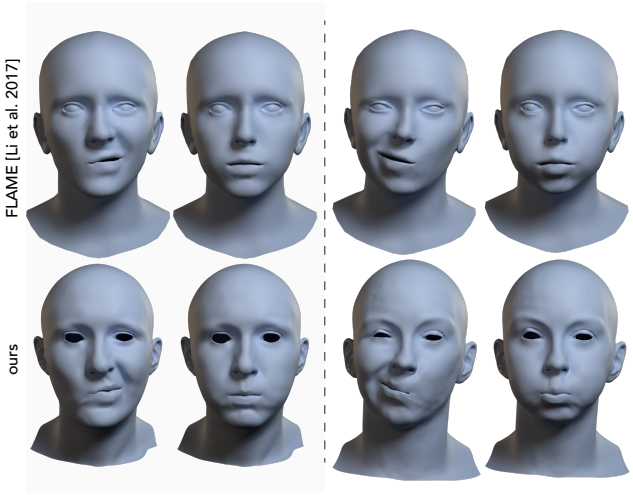

Figure 17. We show 2 expression of 2 different actors fitted by our model and the FLAME model [29]. A generic 3DMM is unable to faithfully capture a particular individuals shape that lies outside of it's shape space.

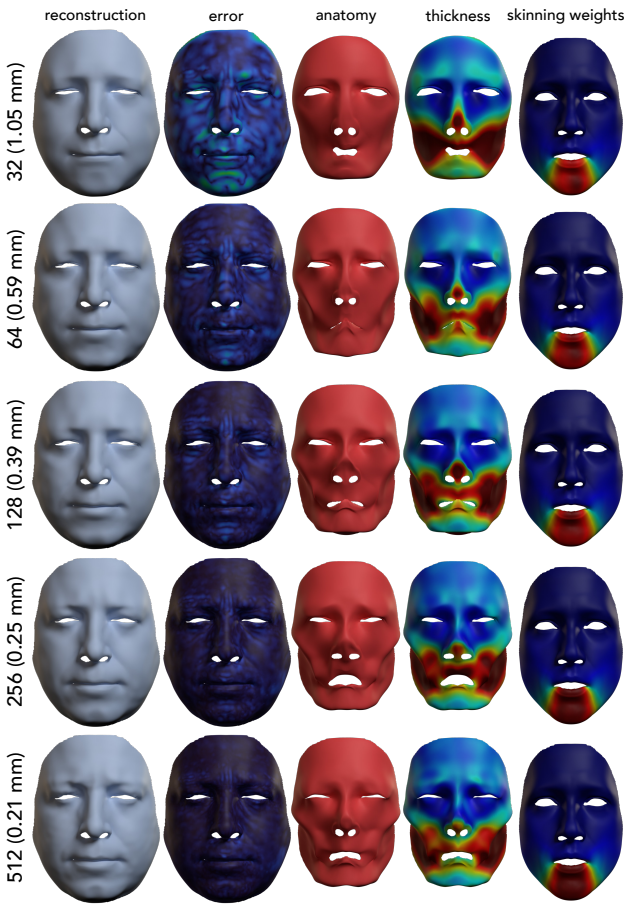

Figure 18. While increasing the size of the hidden layers in our MLPs improved reconstruction performance, it comes at the cost of a larger network that is slower to evaluate. In our work, we used a hidden layer size of 256 neurons which provided a good balance between accuracy and performance.