SUPPLEMENTARY MATERIALS

SUPPL. 1 DATA COLLECTION AND PROCESSING

During data collection, participants were asked to bite into a hydrocolloid material to create maxillary dental impressions. This resulted in a mold which was filled with plaster to obtain a positive cast of the palate and maxillary dentition. Once the casts were cured, they were scanned using a 3D laser scanner (3Shape, Copenhagen, Denmark). Next, the digitized casts were cropped, cleaned, and saved as 3D meshes using '3Shape Ortho Analyzer'. The resulting 3D meshes were then submitted to a spatially dense surface registration (MeshMonk[19]) which consists of the following steps: (1) To initialize the process, a template mesh was roughly aligned to each of the target dental casts, based on five landmarks that were manually indicated at the canines, first molars, and the palatal midline between the first molars. Note that these are not anatomical landmarks and that they were solely used to obtain an initial alignment. Multiple studies have indicated that while small variations of the initializing landmarks can slightly influence the outcome of the registration process, these variations do not carry clinical significance.[34,35] (2) After the first rough alignment, the template was further aligned to the target casts by a scaled rigid registration based on iterative closest point. During this transformation, the template could be translated, rotated, and isotropically scaled. Correspondences were updated using pull-and-push forces and a weighted k-neighbor approach. (3) Following the scaled rigid registration, the template mesh was non-rigidly deformed as to fit the target casts as closely as possible. Similar to step (2), this process was based on a combination of pull-and-push forces and a weighted k-neighbor approach. Additionally, a visco-elastic model ensured that points in proximity of each other moved coherently. (4) In the final step, all target casts were co-aligned and scaled to a common size. Generalized Procrustes analysis was used to eliminate differences in position, orientation, and scale between different dental casts. The template mesh that was used for the registration consists of 8,915 quasi-landmarks and includes the palatal vault, the gums up to the base of the dental cast, and dentition up to the second molars. Third molars were excluded since they often tend to be missing, which could introduce considerable shape variation that we are not interested in. Moreover, the data used in this work was initially collected for a different study (Pittsburgh Orofacial Cleft Study) which focused primarily on shape variation of the palate. Dentition was included to provide guidance for the learning process, but it was not the primary region of interest.

[34] White JD, Ortega-Castrillon A, Virgo C, et al. Sources of variation in the 3dMDface and Vectra H1 3D facial imaging systems. *Scientific Reports*. 2020;10(1):4443. doi:10.1038/s41598-020-61333-3

[35] Verhelst P-J, Matthews H, Verstraete L, et al. Automatic 3D dense phenotyping provides reliable and accurate shape quantification of the human mandible. *Sci Rep*. 2021;11(1):8532. doi:10.1038/s41598-021-88095-w

SUPPL. 2 TRAINING STRATEGY

PRINCIPAL COMPONENT ANALYSIS

The PCA-based SSM is computed using the SVD of the training data. In a first step, the average shape over all training data is subtracted from each dental cast. This allows the model to focus on normal shape variation around the mean. The SVD is a matrix factorization that decomposes data matrix X into uncorrelated variables. However, our dataset contains meshes which are represented by a 3D (Nx8,915x3) tensor of vertices. Therefore, the data tensor should be reshaped into a 2D (Nx26,745) matrix that contains vectorized versions of the meshes before computing the factorization. After reshaping, the matrix X has the following form:

$$X_{N\times 26{,}745} = \begin{bmatrix} x_1^1 & y_1^1 & z_1^1 & x_1^2 & y_1^2 & z_1^2 & \cdots & x_1^{8{,}915} & y_1^{8{,}915} & z_1^{8{,}915} \\ x_2^1 & y_2^1 & z_2^1 & x_2^2 & y_2^2 & z_2^2 & & x_2^{8{,}915} & y_2^{8{,}915} & z_2^{8{,}915} \\ & & \vdots & & & & \ddots & & \vdots & \\ x_N^1 & y_N^1 & z_N^1 & x_N^2 & y_N^2 & z_N^2 & \cdots & x_N^{8{,}915} & y_N^{8{,}915} & z_N^{8{,}915} \end{bmatrix},$$

where superscripts refer to the vertex number and subscripts indicate different individuals. The SVD of X is then defined as $X = U\Sigma V^T$, where Σ is a diagonal matrix containing the singular values s ($s = diag(\Sigma)$), and the columns from U and V are made up from the left and right singular vectors, respectively. The right singular vectors in matrix V are the PCs and can be used to obtain the scores of shape data in the lower dimensional space by multiplication. To reconstruct the original shapes from the scores, the inverse operation can be performed by multiplication with the transpose of V.

AUTOENCODER

The AE-based SSM is a spiral convolutional neural network that directly takes 3D shape data as input. The spiral filters are constructed to consider multiple dimensions at the same time, which means there is no need to reshape the data from a tensor into a matrix before submitting it to the network. Similar to the PCA-based model, the AE network works on data that is normalized by subtracting the average training shape. An AE consists of two parts, an encoder and a decoder whose filter weights are learned simultaneously. During training, the filters were refined iteratively through the backpropagation algorithm[26]. The network was trained for 500 epochs, meaning each training sample was presented to the network 500 times. For the network to successfully learn a given task, a suitable objective function should be defined. We used the reconstruction error, which is defined as the mean absolute error (MAE) between the original shape and its reconstruction. Neural networks are prone to overfitting, this means that the network memorizes the training data rather than learning features that are applicable to unseen data as well. We used two strategies to avoid overfitting: First, the validation set was evaluated alongside the training set during training. If at some point the error of the validation set started increasing while the training error kept decreasing, this was taken as a sign of overfitting and the training was stopped. Second, we applied weight regularization[36], a technique that is designed to modify the learning algorithm such that the model performs better on unseen data. This was done by adding an extra term to the objective function that penalized extreme values in the learned filters and promoted filters to consist of multiple smaller values.

---

[36] Loshchilov I, Hutter F. Decoupled Weight Decay Regularization. In: *International Conference on Learning Representations*. ; 2019. https://openreview.net/forum?id=Bkg6RiCqY7

SINGULAR AUTOENCODER

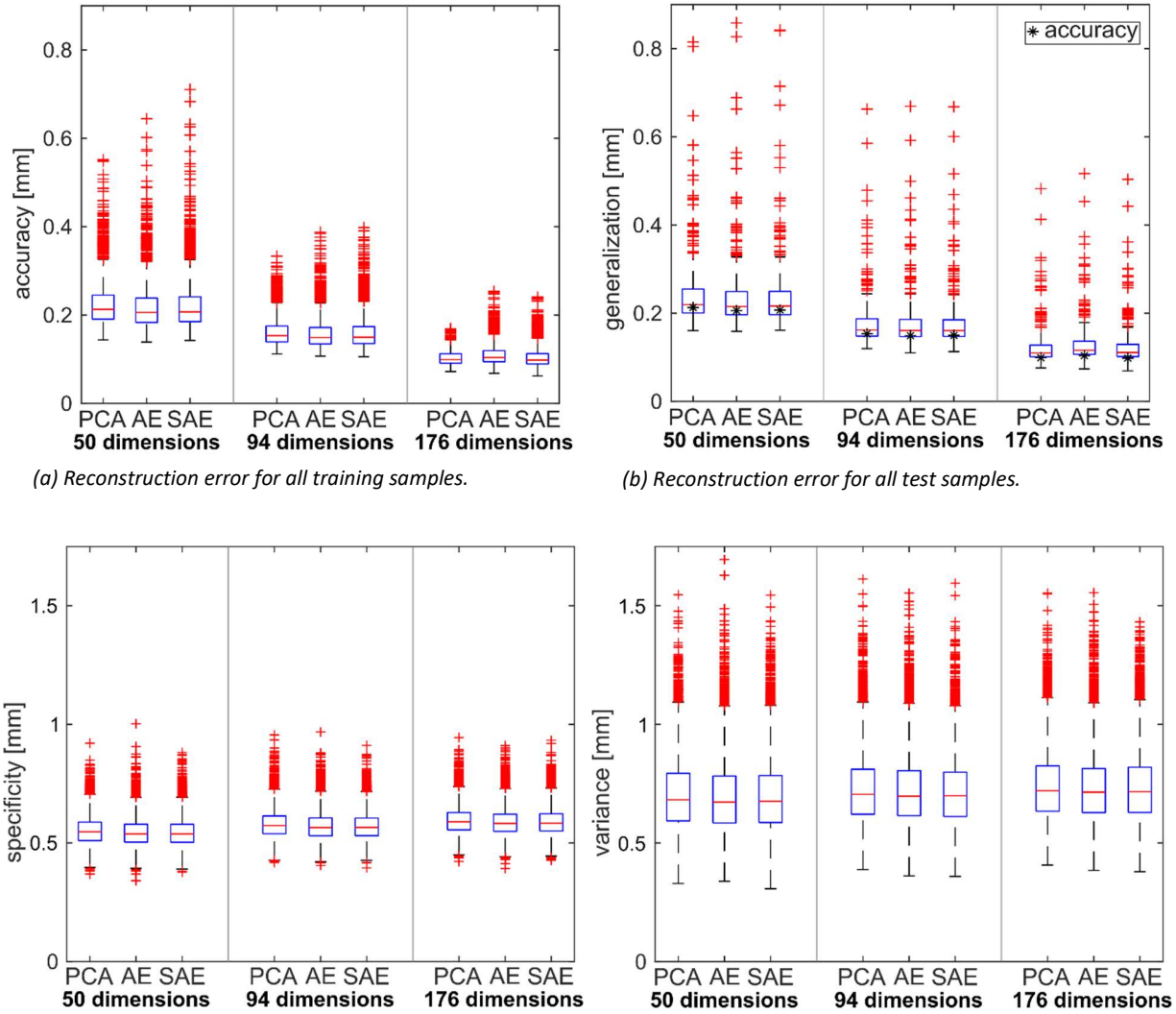
The SAE-based SSM is very similar to the AE-based SSM with only a few small differences in the construction of the network. Therefore, the training strategy of the SAE was largely identical to that of the AE with the only difference being that we started with a pretrained AE network for the SAE. The pretrained AE consisted of the first five layer of the encoder and the last five layers of the decoder. It was trained for 100 epochs and then used to initialize the training for the SAE, which was trained for an additional 500 epochs.

TECHNICAL DETAILS

The SSM based on PCA was constructed in Matlab using a custom toolbox. The AE- and SAE-based SSMs were both constructed in Python using the PyTorch library. The initial AE model was largely based on the publicly available code of the authors of SpiralNet++[24]. We then adapted and extended this code to create the SAE model. All models and Python implementations for the AE and SAE networks are available at https://gitlab.kuleuven.be/u0123700/singular_autoencoder.

SUPPL. 3 INDIVIDUAL SAMPLE RESULTS

Figure S.1 summarizes the results of individual samples for each of the evaluation metrics: accuracy, generalization, specificity, and variance.



(a) Reconstruction error for all training samples.

(b) Reconstruction error for all test samples.

(c) Distance between all synthetic samples and their closest training sample.

(d) Distance between all synthetic samples and the average shape $\bar{x}$.

Figure S.1 The top row shows individual measurements for the reconstruction error, measured by the mean absolute error, of training (a) and test (b) samples. The bottom row summarizes the distances, measured by the mean absolute error, from N=10,000 generated synthetic samples to their most similar training sample (c) and to the average shape $\bar{x}$ (d).

SUPPL. 4 STATISTICAL ANALYSIS

Table S.1 reports the results of a t-test testing whether there are significant differences between the performance of the different models for each of the evaluation metrics. A Bonferroni correction was applied to account for multiple test comparisons. This resulted in a significance threshold of $p < 5.56E\text{-}3$ (i.e., $p < 0.05/9$).

*Table S.1 Statistical results of t-tests comparing the performance of different models for all evaluation metrics. Accuracy and generalization were submitted to a paired t-test with 2,027 and 395 degrees of freedom, respectively. Specificity and variance were submitted to an unpaired t-test with 19,998 degrees of freedom. Significant results (p < 5.56E-3) are indicated with *.*

		ACCURACY		GENERALIZATION		SPECIFICITY		VARIANCE	
		t-value	*p-value*	*t-value*	*p-value*	*t-value*	*p-value*	*t-value*	*p-value*
50	**PCA - SAE**	10,23	6E-24*	3,05	2E-03*	9,52	2E-21*	3,11	2E-03*
	PCA - AE	-30,85	0*	-3,47	6E-04*	-8,89	6E-19*	-4,06	5E-05*
	AE - SAE	17,06	0*	0,56	6E-01	-0,65	5E-01	0,93	4E-01
94	**PCA - SAE**	5,36	9E-08*	-1,63	1E-01	10,97	6E-28*	4,96	7E-07*
	PCA - AE	-14,90	0*	-0,26	8E-01	-10,86	2E-27*	-3,72	2E-04*
	AE - SAE	18,79	0*	3,30	1E-03*	0,04	1E+00	-1,20	2E-01
176	**PCA - SAE**	-5,56	3E-08*	-6,06	3E-09*	6,75	2E-11*	2,98	3E-03*
	PCA - AE	34,90	0*	26,47	0*	-8,33	8E-17*	-3,49	5E-04*
	AE - SAE	-85,58	0*	-37,80	0*	1,59	1E-01	0,55	6E-01