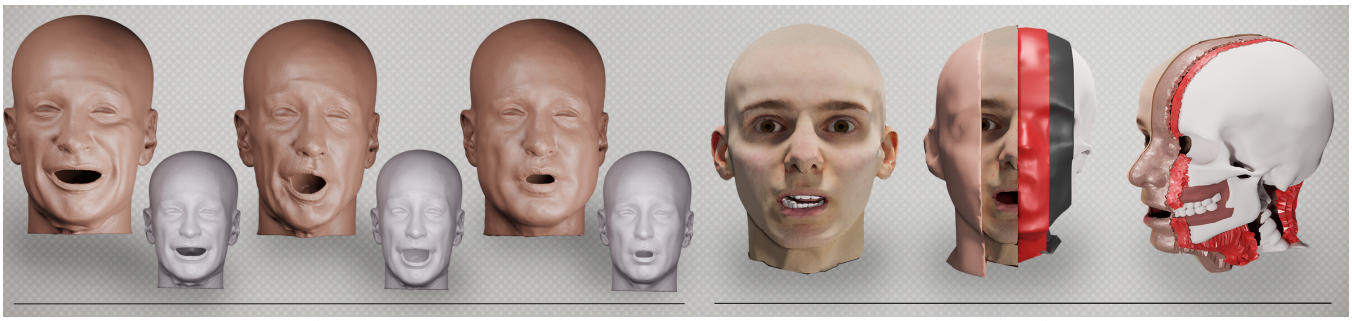
SoftDECA: Computationally Efficient Physics-Based Facial Animations

Nicolas Wagner
nicolas.wagner@tu-dortmund.de
TU Dortmund University
Dortmund, Germany

Ulrich Schwanecke
ulrich.schwanecke@hs-rm.de
RheinMain University of Applied Sciences
Wiesbaden, Germany

Mario Botsch
mario.botsch@tu-dortmund.de
TU Dortmund University
Dortmund, Germany



a) b)

Figure 1: a) SoftDECA (brown) compared to linear blendshapes (gray): More realistic non-linear facial animations (left), biomechanical restrictions like Bell's Palsy (middle), and interactive manipulations like an increase in weight (right) are only a few examples that can be efficiently animated. b) The layered head model that encapsulates the skin, the muscles, and the skull with wraps that builds the foundtion of SoftDECA and for which we present a data-driven fitting algorithm.

ABSTRACT

Facial animation on computationally weak systems is still mostly dependent on linear blendshape models. However, these models suffer from typical artifacts such as loss of volume, self-collisions, or erroneous soft tissue elasticity. In addition, while extensive effort is required to personalize blendshapes, there are limited options to simulate or manipulate physical and anatomical properties once a model has been crafted. Finally, second-order dynamics can only be represented to a limited extent.

For decades, physics-based facial animation has been investigated as an alternative to linear blendshapes but is still cumbersome to deploy and results in high computational cost at runtime. We propose SoftDECA, an approach that provides the benefits of physics-based simulation while being as effortless and fast to use as linear blendshapes. SoftDECA is a novel hypernetwork that efficiently approximates a FEM-based facial simulation while generalizing over the comprehensive DECA model of human identities, facial expressions, and a wide range of material properties that can be locally adjusted without re-training. Along with SoftDECA, we introduce a pipeline for creating the needed high-resolution training data. Part of this pipeline is a novel layered head model that densely positions the biomechanical anatomy within a skin surface while avoiding self-intersections.

CCS CONCEPTS

• **Computing methodologies** → **Physical simulation**; **Neural networks**.

KEYWORDS

Facial Animation, Physics-Based Simulation, Deep Learning

ACM Reference Format:
Nicolas Wagner, Ulrich Schwanecke, and Mario Botsch. 2023. SoftDECA: Computationally Efficient Physics-Based Facial Animations. In *ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '23), November 15–17, 2023, Rennes, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3623264.3624439

1 INTRODUCTION

At present, research in the field of head avatars and facial animation is mainly concerned with obtaining photorealistic results through neural networks [Athar et al. 2022; Cao et al. 2022; Grassal et al. 2022; Zielonka et al. 2023] which can be operated on computationally rich systems. What currently falls short, however, is the inclusion of less capable hardware setups and circumstances in which geometry-based processing must be applicable. For this, various adaptations of linear blendshape models [Lewis et al. 2014] are still the usual means in production. Although linear facial models have been intensively researched and improved over the past decades,

MIG '23, November 15–17, 2023, Rennes, France

ACM ISBN 979-8-4007-0393-5/23/11.
https://doi.org/10.1145/3623264.3624439

there are still known shortcomings like physically implausible distortions, loss of volume, anatomically impossible expressions, missing volumetric elasticity, or self-intersections. Physics-based simulations have been proposed that overcome most artifacts of linear blendshapes and allow for manifold additional functionalities [Barrielle et al. 2016; Choi et al. 2022; Cong 2016; Ichim et al. 2017, 2016; Srinivasan et al. 2021; Yang et al. 2022]. Among them are medical applications such as visualization of weight changes, paralysis, or surgeries but also visual effects like aging, zombifications, gravity changes, and second-order effects. Moreover, it has recently been shown [Yang et al. 2022] that simulations with detailed extracted material information lead to much more realistic facial animations than linear models. The downside of physics-based facial animation models, however, is that these characteristically cause considerable computational overhead, giving rise to a body of literature on acceleration techniques. At this, the focus has been mostly on the evaluation of simulations in either manually constructed [Brandt et al. 2018] or learned subspaces [Holden et al. 2019; Santesteban et al. 2020] as well as on corrective blendshapes [Ichim et al. 2016]. The learned subspace methods [Holden et al. 2019] have proven to be more general and flexible, which is why in SoftSMPL [Santesteban et al. 2020] they have already been successfully applied to full-body animations. Nonetheless, so far there is still no method that transfers these advancements in fast physics-based simulations to facial animations. The principal contribution of this work is closing this gap with a deep learning approach which we call SoftDECA.

SoftDECA is a novel neural network architecture that efficiently animates faces while closely following a dynamic physics-based model. Although our method is universal in the sense that arbitrary physics-based facial animations can be considered, we focus on approximating a combination of state-of-the-art anatomically plausible and volumetric finite element methods (FEM) [Cong and Fedkiw 2019; Cong 2016; Ichim et al. 2017, 2016]. For this, we propose a novel adaption of hypernetworks [Ha et al. 2016] which yields inference times of about 10ms on consumer-grade CPUs and has the same programming interface as standard linear blendshapes. More precisely, we train SoftDECA to be applied as an add-on to arbitrary human blendshape rigs that follow the ARKit system[1].

At the same time, SoftDECA is easily deployable without the need for elaborated personalizations or retraining, as we collect an extensive corpus of training examples. These examples cover a reasonable domain of the targeted FEM and bring together multiple data sources such as CT head scans to reflect the anatomy of heads, 3D head reconstructions in the wild that capture diverse head shapes (DECA [Feng et al. 2021]), and facial expressions in the form of recorded ARKit blendshape weights from dyadic conversational situations. The resulting overall training set facilitates a strong generalization of SoftDECA across human identities, facial expressions, and broad areas of the parameter manifold of the targeted FEM model. In contrast to earlier methods [Holden et al. 2019; Santesteban et al. 2020], the ability to generalize across FEM parameters makes extensive and efficient artistic interventions possible, with SoftDECA even supporting localized material adjustments.

[1]https://developer.apple.com/

As an additional contribution, we present a novel layered head model (LHM) that represents all training instances in a standardized way. Unlike fully or partially tetrahedralized volumetric meshes conventionally used for FEM, the LHM has additional enveloping wraps around bones, muscles, and skin. Based on these wraps, we describe a data-driven fitting procedure that positions muscles and bones within a neutral head while avoiding intersections of the various anatomic structures. A characteristic that was mostly not of concern in previous manually crafted physics-based facial animations but can otherwise lead to numerical instabilities in our automated training data generation approach.

2 RELATED WORK

2.1 Personalized Anatomical Models

Algorithms that create personalized anatomical models can essentially be distinguished according to two paradigms: *heuristic-based* and *data-driven*. Considering heuristic-based approaches, Anatomy Transfer [Ali-Hamadi et al. 2013] applies a space warp to a template anatomical structure to fit a target skin surface. The skull and other bones are only deformed by an affine transformation. A similar idea is proposed by Gilles et al. [2010]. While they also implement a statistical validation of bone shapes, the statistics are collected from artificially deformed bones. In [Ichim et al. 2016; Kadleček et al. 2016], an inverse physics simulation was used to reconstruct anatomical structures from multiple 3D expression scans. Saito et al. [2015] simulate the growth of soft tissue, muscles, and bones. A musculoskeletal biomechanical model is fitted from sparse measurements in [Schleicher et al. 2021] but not qualitatively evaluated.

There are only a few data-driven approaches because combined data sets of surface scans and CT, or CT and DXA images are hard to obtain for various reasons (e.g. data privacy or unnecessary radiation exposure). The recent work OSSO [Keller et al. 2022] predicts full body skeletons from 2000 DXA images that do not carry precise 3D information. Further, bones are positioned within a body by predicting only three anchor points per bone group and not avoiding intersections between skin and skull. A model that prevents skin-skull intersections and also considers muscles is based on fitting encapsulating wraps instead of the anatomy itself [Komaritzan et al. 2021]. However, no accurate algorithm based on medical imaging but a BMI (body mass index) regressor [Maalin et al. 2021] is used to position the wraps. A much more accurate, pure face model, was developed by Achenbach et al. [2018]. Here, CT scans are combined with optical scans by a multilinear model (MLM) which can map from skulls to faces and vice versa. As before, no self-intersections are prevented and only bones are fitted. Building on the data from [Achenbach et al. 2018] and following the idea of a layered body model [Komaritzan et al. 2021], we create a statistical layered head model including musculature that avoids self-intersections.

2.2 Physics-Based Facial Animation

A variety of techniques for animating faces have been developed in the past [Bradley et al. 2010; Ichim et al. 2015; Parke 1991; Zhang et al. 2008]. Data-driven models [Ichim et al. 2016; Lewis et al. 2014, 2005], which have recently been significantly improved by deep learning [Athar et al. 2022; Cao et al. 2022; Feng et al. 2021; Garbin

et al. 2022; Song et al. 2020; Zheng et al. 2022], are certainly dominant. Due to their simplicity and speed, linear blendshapes [Lewis et al. 2014] are still most commonly used in demanding applications and whenever no computationally rich hardware is available. Physics-based models have been developed for a long time [Sifakis et al. 2005] and avoid artifacts like implausible contortions and self-intersections, but due to their complexity and computational effort, they are rarely used. The pioneering work of Sifakis et al. [2005] is the first fully physics-based facial animation. The simulation is conducted on a personalized tetrahedron mesh, which can only be of a limited resolution due to a necessary dense optimization problem. With Phace [Ichim et al. 2017], this problem was overcome by an improved physics simulation. An art-directed muscle model [Bao et al. 2019; Cong and Fedkiw 2019; Cong 2016] additionally represents muscles as B-splines and allows control of expressions via trajectories of spline control points. A solely inverse model for determining the physical properties of faces was proposed in [Kadleček and Kavan 2019].

Hybrid approaches add surface-based physics to linear blendshapes for more detailed facial expressions [Barrielle et al. 2016; Bickel et al. 2008; Choi et al. 2022; Kozlov et al. 2017]. However, by construction, they can not model volumetric effects. With volumetric blendshapes [Ichim et al. 2016], a hybrid approach has been presented that combines the structure of linear blendshapes with volumetric physical and anatomical plausibility but can only achieve real-time performance through personalized corrective blendshapes.

Considering soft bodies in general, deep learning approaches have been investigated to approximate physics-based simulations. For instance, in [Casas and Otaduy 2018; Santesteban et al. 2020] the SMPL (Skinned Multi-Person Linear Model) proposed in [Loper et al. 2015] was extended with secondary motion. Recently, [Choi et al. 2022; Srinivasan et al. 2021; Yang et al. 2022] developed methods to learn the particular physical properties of objects and faces. However, these approaches must be retrained for unseen identities and are slow in inference. A fast and general approach for learning physics-based simulations is introduced in [Holden et al. 2019]. Unfortunately, they focused on reflecting the dynamics of single objects with limited complexity. We present a real-time capable deep learning approach to physics-based facial animations that does not need to be retrained and maintains the control structure of standard linear blendshapes. Additionally, none of the previously described deep learning methods tackle the challenging creation of facial training data, which we also address in this work.

3 METHOD

The foundation of the SoftDECA animation system is a novel layered head representation (Section 3.1). Starting from there, we design a FEM-based facial animation system (Sections 3.2 & 3.3) and demonstrate how to distill it into a defining dataset (Section 3.4). With this dataset, we train a newly designed hypernetwork (Section 3.5) as a real-time capable approximation of the animation system.

Figure 2: All components of the layered head model template $\mathcal{T}$. Skin $S_\mathcal{T}$, skin wrap $\hat{S}_\mathcal{T}$, muscles $M_\mathcal{T}$, muscles wrap $\hat{M}_\mathcal{T}$, skull $B_\mathcal{T}$, and the skull wrap $\hat{B}_\mathcal{T}$.

3.1 Layered Head Model

3.1.1 Structure. We represent a head $\mathcal{H} = \rho_\mathcal{H}(\mathcal{T})$ with neutral expression through a component-wise transformation $\rho_\mathcal{H}$ of a layered head model template

$$\mathcal{T} = \left(S_\mathcal{T},\ B_\mathcal{T},\ M_\mathcal{T},\ \hat{S}_\mathcal{T},\ \hat{B}_\mathcal{T},\ \hat{M}_\mathcal{T}\right), \tag{1}$$

that consists of six triangle meshes. $S_\mathcal{T}$ describes the skin surface including the eyes, the mouth cavity, and the tongue, $B_\mathcal{T}$ the surface of all skull bones and teeth, $M_\mathcal{T}$ the surface of all muscles and the cartilages of the ears and nose. $\hat{S}_\mathcal{T}$ is the skin wrap, i.e. a closed wrap enveloping $S_\mathcal{T}$, $\hat{B}_\mathcal{T}$ the skull wrap that envelopes $B_\mathcal{T}$, and $\hat{M}_\mathcal{T}$ the muscle wrap that envelopes $M_\mathcal{T}$. Other anatomical structures are omitted for simplicity. The template structures $S_\mathcal{T}$, $B_\mathcal{T}$, and $M_\mathcal{T}$ were designed by an experienced digital artist. The skin, skull, and muscle wraps $\hat{S}_\mathcal{T}$, $\hat{B}_\mathcal{T}$, and $\hat{M}_\mathcal{T}$ have the same triangulation and were generated by shrink-wrapping a sphere as close as possible to the corresponding surfaces without intersections. The complete template is shown in Figure 2.

Due to the shared triangulation, the wraps of the LHM also define a soft tissue tet mesh $\mathbb{S}_\mathcal{T}$ (i.e. between the skin and the muscle wraps) and a muscle tissue tet mesh $\mathbb{M}_\mathcal{T}$ (i.e. between the muscle and the skull wraps). For this, each triangle prism that can be spanned between corresponding wrap faces is canonically split into three tets. The complexities of all template components are given in the supp. material. In the following, we will state the number of vertices of a mesh as $|\cdot|_v$ and the number of faces as $|\cdot|_f$.

3.1.2 Fitting. Later on, creating training data requires finding

$$\left(S,\ B,\ M,\ \hat{S},\ \hat{B},\ \hat{M}\right) = \rho_\mathcal{H}(\mathcal{T}) \tag{2}$$

when only the skin surface S of the head $\mathcal{H}$ is known. To this end, we rely on a hybrid approach that positions the skull in a data-driven manner while the remaining template components are fitted by heuristics that ensure anatomic plausibility and avoid self-intersections.

As the first of the remaining template meshes, we fit the skin wrap by setting

$$\hat{S} = \mathrm{rbf}_{S_\mathcal{T} \to S}(\hat{S}_\mathcal{T}). \tag{3}$$

The RBF function is a space warp based on triharmonic radial basis functions [Botsch and Kobbelt 2005] that is calculated from the

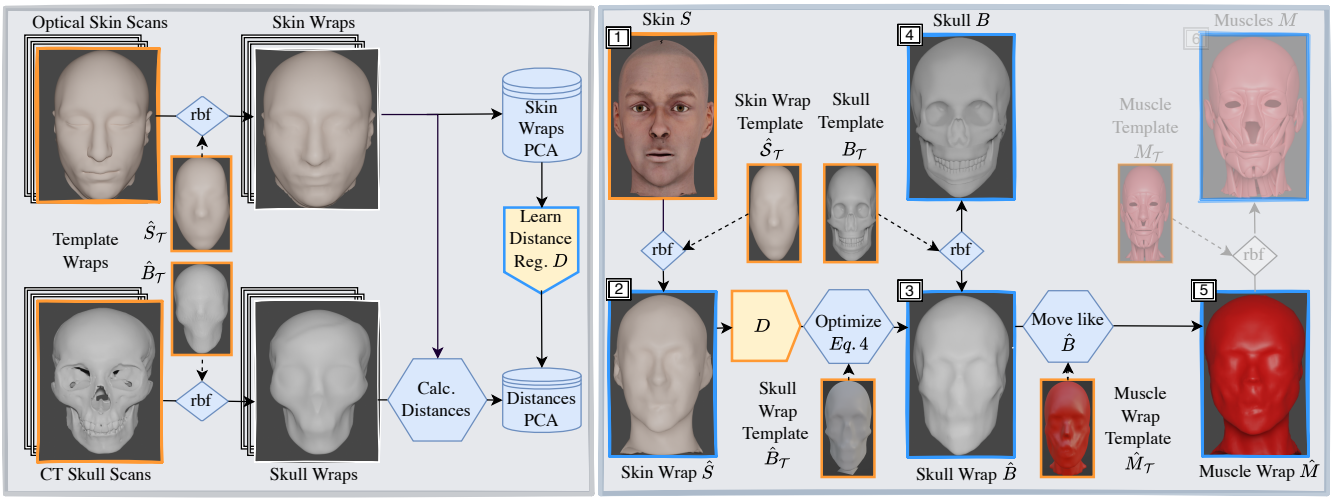


Figure 3: a) The training scheme of the skin to skull wrap distances regressor D. b) Procedural overview of the layered head model fitting algorithm. Orange frames indicate input, blue frames output. The enumeration reflects the fitting order. Step 6 is shown only for the sake of completeness.

template skin surface $S_\mathcal{T}$ to the target S and subsequently applied to the template skin wrap $\hat{S}_\mathcal{T}$. By the construction of RBFs, the skin wrap will be warped semantically consistent and stick close to the targeted skin surface.

Next, we fit the skull wrap $\hat{B}$ by invoking a linear regressor D that predicts the distances from the vertices of $\hat{S}$ to the corresponding vertices of $\hat{B}$ and subsequently minimizing with projective dynamics [Bouaziz et al. 2014]

$$\begin{aligned}\arg\min_X w_{\text{rect}} E_{\text{rect}}\left(X,\ \hat{S}_\mathcal{T}\right) + w_{\text{dist}_2} E_{\text{dist}_2}\left(X,\ \hat{S},\ D\left(\hat{S}\right)\right) \\ + w_{\text{curv}} E_{\text{curv}}\left(X,\ \hat{B}_\mathcal{T}\right).\end{aligned} \tag{4}$$

Here, E_{dist_2} ensures that the predicted distances are adhered to, E_{curv} is a curvature regularization of the skull wrap, and E_{rect} avoids shearing between corresponding skin and skull wrap faces. The distances are set to a minimum value if they fall below a threshold, thus, avoiding skin-skull intersections. To ease the flow of reading, we give formal descriptions of the energy components in the supp. material. The optimization is initialized with $X = \hat{S} - D(S) \cdot n(\hat{S})$ where $n(\hat{S})$ are area-weighted vertex normals. D is trained on the dataset of [Gietzen et al. 2019] (SKULLS) that relates CT skull measurements to optical skin surface scans. In Figure 3 a) the linear regressor training is depicted.

The muscle wrap $\hat{M}$ is fitted by positioning its vertices at the same absolute distances between the corresponding skin and skull wrap vertices as in the template, and only passing on ten percent of the relative distance changes compared to the template. This approach assumes that the muscle mass in the facial area is only moderately affected by body weight and skull size.

The skull mesh is placed by setting

$$B = \text{rbf}_{\hat{B}_\mathcal{T} \to \hat{B}}(B_\mathcal{T}). \tag{5}$$

The properties of the RBF space warp ensure that the skull mesh remains within the skull wrap if the wrap is of sufficient resolution. The muscle mesh could be placed in a similar fashion but is not needed in our pipeline any further.

Finally, the soft and muscle tissue tet meshes $\mathbb{S}$ and $\mathbb{M}$ can be constructed as described before. On average, the complete fitting pipeline takes about 500ms on an AMD Threadripper Pro 3995wx processor. Figure 3b) visualizes the overall fitting process.

3.2 SoftDECA Animation System

Building on the LHM representation, we now introduce the SoftDECA animation system. For this, the classical concept of linear blendshapes is reviewed first. Thereupon, the dynamic physics-based facial simulation system which is at the core of SoftDECA is derived.

For a specific head, a linear blendshape model consists of n surface blendshapes

$$\{S^i\}_{i=1}^n \tag{6}$$

which animate an unknown facial expression S_t as a linear combination

$$S_t = \sum_{i=1}^N \mathbf{w}_t^i S^i, \tag{7}$$

where the blending weights $\mathbf{w}_t$ determine the share of each blendshape in the expression at frame t.

To achieve the same animation with a physical model ϕ, one typically differentiates between forward and inverse methods. Without loss of the generality, we consider the inverse method in the following. Here, the expression S_t is converted into the (in the Euclidean sense) closest $\phi-$plausible solution by $\phi^\dagger$ to

$$T_t = \phi^\dagger(S_t,\ \mathbf{p}), \tag{8}$$

where $\mathbf{p}$ is a vector of material and simulation parameters on which ϕ depends. For including second-order effects as well, Equation (8) expands to

$$T_t = \phi^\dagger(\gamma\, S_t + 2\alpha\, T_{t-1} - \beta\, T_{t-2},\ \mathbf{p}). \tag{9}$$

The SoftDECA animation system operates in the same manner, but the right-hand side is approximated by a computationally efficient neural network f.

Next, we will describe our realization of $\phi^\dagger$ and how to create representative examples. Nonetheless, please note that SoftDECA is not restricted to a particular realization of $\phi^\dagger$.

3.3 Physics-Based Simulations

We implement anatomically plausible inverse physics $\phi^\dagger$ as a projective dynamics energy $E_{\phi^\dagger}$. At this, state-of-the-art FEM models [Cong 2016; Ichim et al. 2017; Kadleček and Kavan 2019] are merged by applying separate terms for soft tissue, muscle tissue, the skin, the skull, and auxiliary components.

3.3.1 Energy. Considering the soft tissue $\mathbb{S}$, we closely follow the model of [Ichim et al. 2017] and impose

$$E_{\mathbb{S}} = w_{\text{vol}} \sum_{\text{t}\in\mathbb{S}} E_{\text{vol}}(\text{t}) + w_{\text{str}} \sum_{\text{t}\in\mathbb{S}} \mathbb{1}_{\sigma_{F(\text{t})}>\epsilon} E_{\text{str}}(\text{t}), \tag{10}$$

which for each tet t penalizes change of volume and strain, respectively. Strain is only accounted for if the largest eigenvalue $\sigma_{F(\text{t})}$ of the stretching component of the deformation gradient $F(\text{t}) \in \mathbb{R}^{3\times3}$ grows beyond ϵ.

To reflect the biological structure of the skin, we additionally formulate a dedicated strain energy

$$E_S = \sum_{\text{t}\in S} E_{\text{str}}(\text{t}) \tag{11}$$

on each triangle t of the skin which, to the best of our knowledge, has not been done before.

For the muscle tets $\mathbb{M}$, we follow Kadleček et al. [2019] that capturing fiber directions for tetrahedralized muscles is in general too restrictive. Hence, only a volume-preservation term

$$E_{\mathbb{M}} = w_{\text{vol}} \sum_{\text{t}\in\mathbb{M}} E_{\text{vol}}(\text{t}) \tag{12}$$

is applied for each tet in $\mathbb{M}$.

The skull is not tetrahedralized as it is assumed to be non-deformable even though it is rigidly movable. The non-deformability of the skull is represented by

$$E_B = \sum_{\text{t}\in B} E_{\text{str}}(\text{t}) + \sum_{\text{x}\in B} E_{\text{curv}}(\text{x}, B), \tag{13}$$

i.e. a strain E_{str} on the triangles t and mean curvature regularization on the vertices x of the skull B. We do not model the non-deformability as a rigidity constraint due to the significantly higher computational burden.

To connect the muscle tets as well as the eyes to the skull, connecting tets are introduced similar to the sliding constraints in [Ichim et al. 2017]. For the muscle tets, each skull vertex connects to the closest three vertices in $\mathbb{M}$ to form a connecting tet. For the eyes, connecting tets are formed by connecting each eye vertex to the three closest vertices in B. On these connecting tets, the energy E_{con} with the same constraints as in Equation (10) is imposed. By this design, the jaw and the cranium are moved independently from each other through muscle activations but the eyes remain rigid and move only with the cranium.

Finally, the energy

$$E_{\text{inv}} = \sum_{\text{x}\in S} E_{\text{tar}}(\text{x},\ S_t) \tag{14}$$

of soft Dirichlet constraints is added, attracting the skin surface S vertices to the targeted expression S_t.

The weighted sum of the aforementioned energies gives the total energy

$$\begin{aligned} E_{\phi^\dagger} = \ & w_{\mathbb{S}} E_{\mathbb{S}} + w_{\mathbb{M}} E_{\mathbb{M}} + w_{\text{B}} E_B + w_{\text{mstr}} E_{\text{mstr}} \\ & + w_{\text{S}} E_S + w_{\text{con}} E_{\text{con}} + w_{\text{inv}} E_{\text{inv}} \end{aligned} \tag{15}$$

of the inverse model $\phi^\dagger$. Altogether, $\phi^\dagger$ results in an expression T_t that in a Euclidean sense is close to the target S_t but is plausible w.r.t. the imposed constraints.

3.3.2 Collisions. Finally, self-intersections are resolved between colliding lips or teeth in a subsequent projective dynamics update as in [Komaritzan and Botsch 2018].

3.3.3 Parameters. The construction of $\phi^\dagger$ also implies parts of the parameter vector $\mathbf{p}$. As such, the dynamics parameters α, β, γ, weights w_* of all the constraints, but also other attributes of the constraints can be considered. For example, the target volume in E_{vol} or scaling factors of the skull bones. Additionally, we include constant external forces like gravity strength and direction into $\mathbf{p}$. An overview of all parameters we use and the corresponding value ranges is given in the supp. material.

3.4 Training Data

By the definition of the animation system in Equation (9), a representative training dataset $\mathcal{D}$ must consist of examples that relate diverse facial expressions created via linear blendshapes to the corresponding surfaces that conform ϕ. Further, to capture dynamic effects, the exemplary facial expressions have to form reasonable sequences. This dataset must also cover a variety of distinct head shapes as well as simulation parameters.

In the following, we describe a pipeline for creating instances of such a dataset, which can be roughly divided into six high-level steps.

(1) We start by randomly drawing a neutral skin surface S from DECA [Feng et al. 2021], a comprehensive high-resolution face model. More specifically, we randomly draw an image from the Flickr-Faces-HQ [Karras et al. 2019] dataset and let DECA determine the corresponding neutral head shape as well as a latent representation $\mathbf{h}$.
(2) Next, the template LHM $\mathcal{T}$ is aligned with the skin surface S as described in Section 3.1.
(3) In the third step, deformation transfer [Botsch et al. 2006] is used to transfer ARKit surface-based blendshapes to S.
(4) Subsequently, we create an expression sequence $\mathbf{S} = (S_t)_{t=0}^m$ of length $m+1$ by applying a sequence of blendshape weights $\mathbf{w} = (\mathbf{w}_t)_{t=0}^m$. The blendshape weights are obtained from 8 around 10 minutes long dyadic conversations recorded with a custom iOS app.
(5) As the final step before the ϕ-plausible counterpart of $\mathbf{S}$ can be generated, simulation parameters have to be sampled on a proper domain. For continuous parameters, we expect the

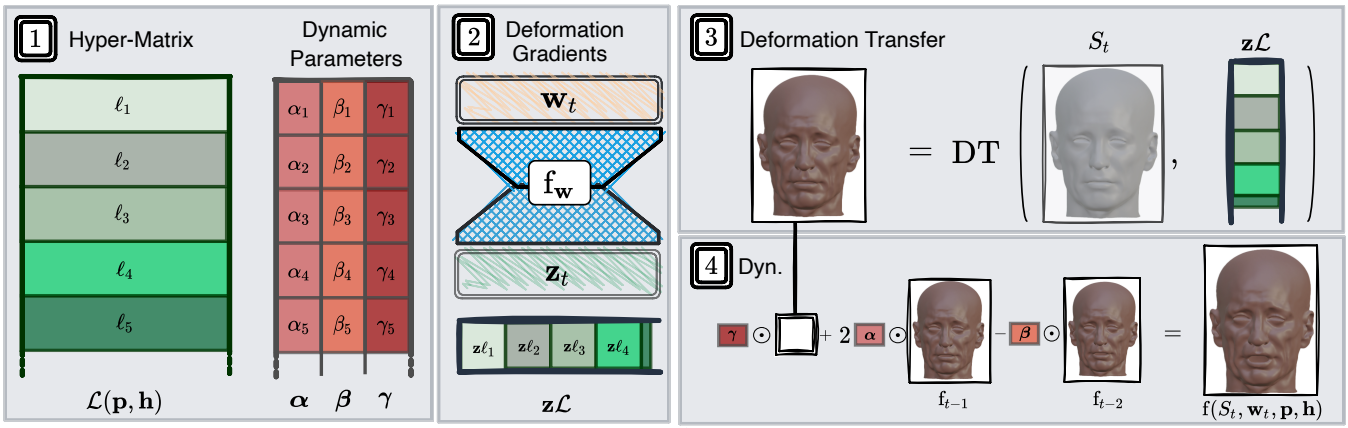


Figure 4: An overview of the SoftDECA facial animation. In Step 1), the hyper-tensor and the dynamic parameters are determined once for an animation. Subsequently, steps 2-4 are repeatedly evaluated per frame. In Step 2), per-face deformation gradients are calculated which are applied in Step 3) to form a facial expression. In Step 4), dynamic effects are added.

user to specify lower and upper bounds beforehand. Subsequently, for each parameter in $\mathbf{p}$, we independently sample a value between the respective bounds with uniform distribution. Discrete parameters are handled in the same way but without respecting particular constraints.

(6) Finally, $\mathbf{T} = (\phi^\dagger(S_t))_{t=0}^m$ is computed and $(\mathbf{T}, \mathbf{S}, \mathbf{w}, \mathbf{p}, \mathbf{h})$ is added to $\mathcal{D}$. Evaluating one time step takes approximately 10 seconds on an AMD Threadripper Pro 3995wx.

3.5 Hypernetwork

3.5.1 Architecture & Training. Having training data, we can now design a computationally efficient neural network f to approximate the physics-based simulation from Equation 9. Irrespective of a particular architecture, the training goal implied by $\mathcal{D}$ is to optimize on each frame

$$\min_{\mathrm{f}} \sum_{(\mathbf{T},\mathbf{S},\mathbf{w},\mathbf{p},\mathbf{h})\in\mathcal{D}} \sum_{t=0}^{m} \|T_t - \mathrm{f}(S_t, \mathbf{w}_t, \mathbf{p}, \mathbf{h})\|_2 \,. \tag{16}$$

In words, f is trained to approximate the ϕ-conformal expressions from the the linearly blended expressions S_t, the blending weights $\mathbf{w}_t$, simulation parameters $\mathbf{p}$, and the head descriptions $\mathbf{h}$. Hence, leaving out dynamic effects to begin with, the probably most naive approach would be to learn f to directly predict vertex positions. However, this would not allow the usage of personalized blendshapes at inference time that have not been used in the curation of $\mathcal{D}$. Therefore, we separate f into two high-level components

$$\mathrm{f}(S_t, \mathbf{w}_t, \mathbf{p}, \mathbf{h}) = \mathrm{DT}(S_t, \mathrm{f}_{DG}(\mathbf{w}_t, \mathbf{p}, \mathbf{h})) \,, \tag{17}$$

where DT is a deformation transfer function as in [Sumner and Popović 2004] that applies 3×3 per-face deformation gradients (DGs) predicted by $\mathrm{f}_{DG}(\mathbf{w}_t, \mathbf{p}, \mathbf{h}) \in \mathbb{R}^{|S|_f \times 9}$ to the linearly blended S_t. By doing so, f can also be applied to a facial expression S_t which has been formed by unseen personalized blendshapes while still achieving close approximations of $\phi^\dagger$. Fortunately, the evaluation of DT is not more than efficiently finding a solution to a pre-factorized linear equation system.

To implement the DG prediction network f_{DG}, we evaluated multiple network architectures such as set transformers [Lee et al. 2019], convolutional networks on geometry images, graph neural networks [Scarselli et al. 2008], or implicit architectures [Mildenhall et al. 2021], but all have exhibited substantially slower inference speeds while reaching a similar accuracy as a multi-layer perceptron (MLP). Nevertheless, a plain MLP does not discriminate between inputs that change per frame t and inputs that have to be computed only once. Therefore, we propose an adaptation of a hypernetwork MLP [Ha et al. 2016] to implement f_{DG} in which the conditioning of f_{DG} with respect to the simulation parameters as well as the DECA identity is done by manipulating network parameters. Formally, we implement

$$\mathrm{f}_{DG}(\mathbf{w}_t, \mathbf{p}, \mathbf{h}) = \mathbf{z}_t \mathcal{L}(\mathbf{p}, \mathbf{h}), \tag{18}$$

where $\mathcal{L}(\mathbf{p}, \mathbf{h}) \in \mathbb{R}^{32\times|S|_f\times 9}$ returns a tensor that only has to be calculated once for all frames and $\mathbf{z}_t = \mathrm{f}_\mathbf{w}(\mathbf{w}_t) \in \mathbb{R}^{32}$ is the result of a small standard MLP that processes the blending weights at every frame t. Each matrix $\ell_i \in \mathbb{R}^{32\times 9}$ in $\mathcal{L}(\mathbf{p}, \mathbf{h})$ corresponds to a face in S and the entries are calculated as

$$\ell_i = \mathrm{f}_\mathbf{ph}(\mathbf{p}, \mathbf{h}, \pi(i)) \,. \tag{19}$$

Again, $\mathrm{f}_\mathbf{ph}$ is a small MLP and π is a trainable positional encoding. Please consult the supp. material for detailed dimensions of all networks and see Figure 4 for a structural overview of f.

3.5.2 Localization. The architecture described above offers extensive possibilities for artistic user interventions at inference time. For instance, different simulation parameters $\mathbf{p}_i$ can be used per face i by changing Equation (19) to

$$\ell_i = \mathrm{f}_\mathbf{ph}(\mathbf{p}_i, \mathbf{h}, \pi(i)) \,, \tag{20}$$

which enables a localized application of different material models. The DT function ensures that the models are smoothly combined.

3.5.3 Dynamics. Given that locally differing simulation parameters are not reflected in the training data, existing approaches to integrate dynamics in deep learning [Holden et al. 2019; Santesteban et al. 2020], cannot be adopted. Therefore, we again use the hypernetwork concept to achieve a piecewise-linear dynamics

approximation. More precisely, we recursively extend f to

$$\begin{aligned} \mathrm{f}(S_t, \mathbf{w}_t, \mathbf{p}, \mathbf{h}) = \quad & \boldsymbol{\gamma} \odot \mathrm{DT}(S_t, \mathrm{f}_{DG}(\mathbf{w}_t, \mathbf{p}, \mathbf{h})) \\ & + 2\boldsymbol{\alpha} \odot \mathrm{f}(S_{t-1}, \mathbf{w}_{t-1}, \mathbf{p}, \mathbf{h}) \\ & - \boldsymbol{\beta} \odot \mathrm{f}(S_{t-2}, \mathbf{w}_{t-2}, \mathbf{p}, \mathbf{h}), \end{aligned} \tag{21}$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{R}^{32\times|S|_v}$ contain per-vertex dynamics parameters. The first row of Equation (21) is the same as in Equation (17) but the second and third rows allow for dependencies on the previous two frames. Each entry of $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ is calculated as in Equation (20) but with dedicated MLPs $\mathrm{f}_\alpha, \mathrm{f}_\beta, \mathrm{f}_\gamma$. As a result, $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ are again not time-dependent and only have to be calculated once.

4 EXPERIMENTS

Before demonstrating the accuracy and efficiency of SoftDECA (Section 4.2), we first evaluate the fitting precision of the LHM (Section 4.1).

4.1 LHM Fitting

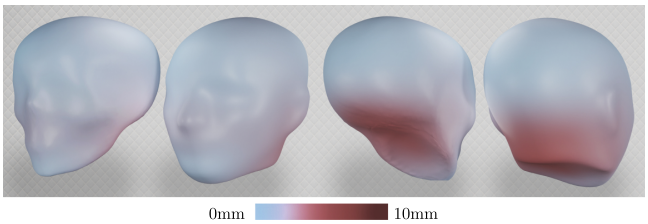


Figure 5: The per-vertex mean L2-error of the LHM fitting.

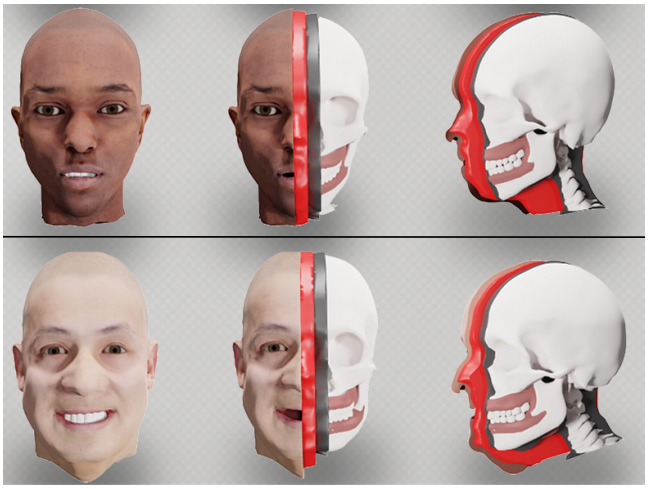


Figure 6: Exemplary fits of the LHM components skull wrap, muscle wrap, and skull.

The fitting of the LHM is mainly composed of the data-driven positioning of the skull wrap and the subsequent heuristic fitting of the muscle wrap. We evaluate the crucial fitting of the skull wrap with the open-source CT SKULLS [Gietzen et al. 2019] dataset. Since this dataset consists of 43 instances only, a leave-one-out validation is performed in which the vertex-wise L2 errors are measured. Earlier methods that position the skull within the head, mainly use sparse soft tissue statistics measured in normal directions starting from very few points on the skull [Beeler and Bradley 2014; Ichim et al. 2016]. We compare our approach to the multilinear model of Achenbach et al. [2018; 2019], who have shown a more robust and precise positioning by capturing dense soft tissue statistics as radii of spheres surrounding the skull.

Both models cannot achieve a medical-grade positioning with errors between approximately 2 mm and 4 mm. The MLM achieves a higher precision with a mean error of 1.98 mm than our approach that dispositions the skull by 3.83 mm on average. However, the MLM cannot prevent collisions that might crash physics-based simulations. Also, our fitting algorithm produces large errors only in regions that are of less importance for facial simulations as can be seen in Figure 5. The errors are predominately distributed in the back area of the skull since here the rectangular constraints of our fitting procedure can presumably no longer be aligned well with the skin wrap. Figure 6 displays fitting examples.

4.2 SoftDECA

4.2.1 Dataset & Training. To train and evaluate f, we assemble a dataset of 500k training and test instances by using the pipeline from Section 3.4. The parallelized dataset creation took five days and required one terabyte of storage. To match the uneven sizes of the parameter spaces, 75% of the produced data is static data in which all but the dynamic parameters α, β, γ are sampled and only the remaining 25% of the data is simulated dynamically. As a result, 6250 dynamic sequences have been generated, each of which has a length of 16 while the static examples consist of only one frame per example. To initialize the dynamic sequences with a reasonable velocity, a longer sequence of length 2048 has been simulated with fixed dynamics parameters a priori. For each dynamic sequence, a random observed velocity of the long sequence is drawn as the initialization. The dataset is split in 90% for training and 10% for testing while neither the same identity nor the same simulation parameters nor the same facial expression occurs in both.

For training, the Adam optimizer performs 200k update steps with a learning rate of 0.0001. The learning rate is linearly decreased to 0.00005 over the course of training and a batch size of 128 is applied. In total, the training specifications result in an approximate runtime of 8 hours on an NVIDIA A6000. The comparatively short training time can straightforwardly be explained by the efficient network design and the less noisy training data than usually encountered for instance in image-based deep learning. We quantitatively evaluate SoftDECA based on the L2 reconstruction error with respect to the targeted physics-based simulation and the computational runtimes. Besides, we compare it against the Subspace Neural Physics (SNP) [Holden et al. 2019] and the SoftSMPL [Santesteban et al. 2020] architectures adapted to facial simulations. These are, to the best of our knowledge, state-of-the-art methods for fast approximations of physics-based simulations. An overview of all results is given in Table 1. The stated runtimes are averages of ten runs measured on a consumer-grade Intel i5 12600K processor. All implementations rely on PyTorch[2].

[2]https://pytorch.org

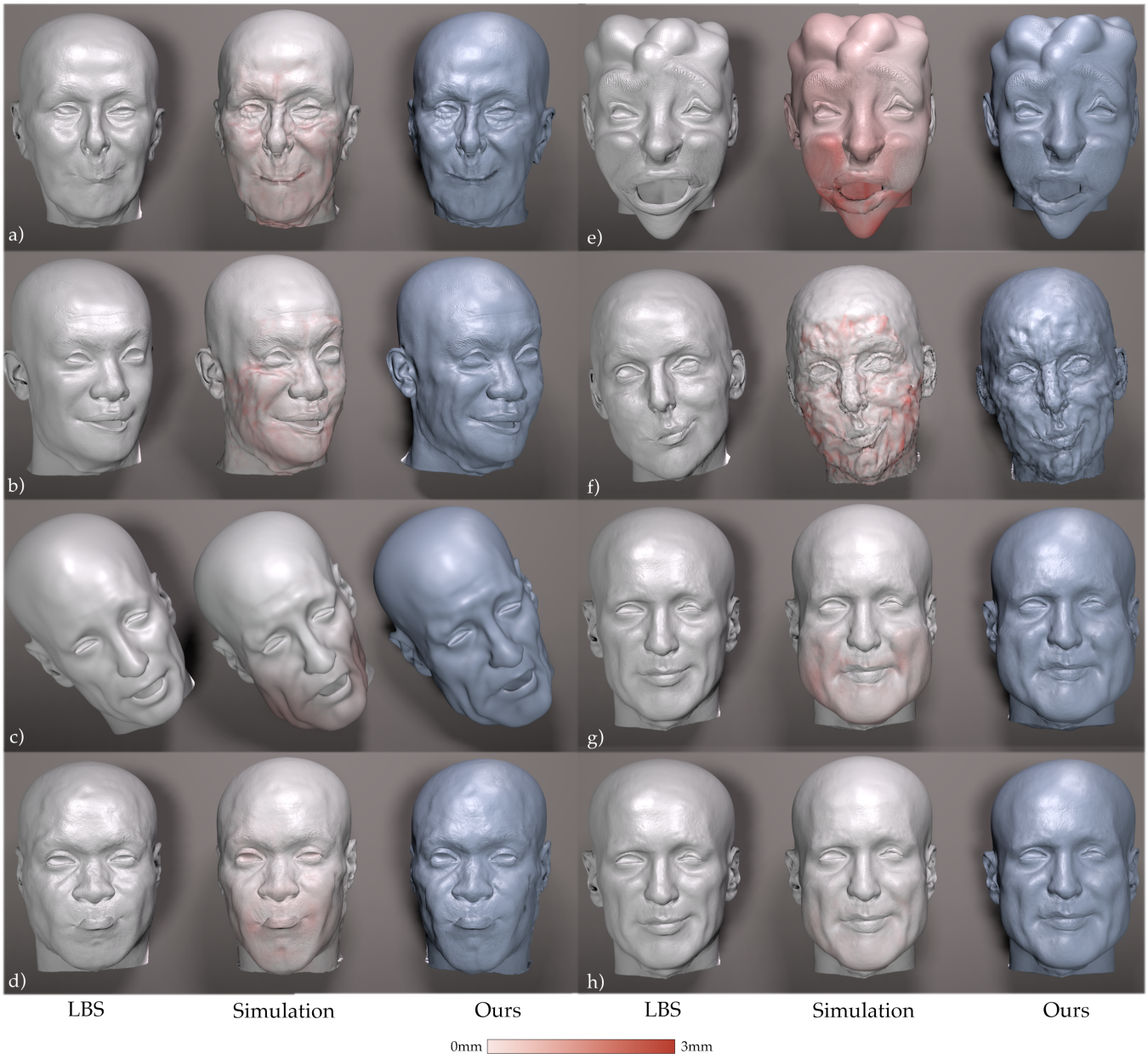


Figure 7: Exemplary results of SoftDECA in comparison to the targeted physics-based facial simulation as well as the inputted linear blendshape expressions. Reconstruction errors are plotted on the simulated expressions.

Table 1: SoftDECA test results in comparison to adapted SNP [Holden et al. 2019] and SoftSMPL [Santesteban et al. 2020] architectures as well as ablations. The runtimes are averages measured on a consumer-grade Intel i5 12600K processor. External refers to the 3Dscanstore dataset. Small and large correspond to the size of the inspected MLP.

Model	Ours			SoftSMPL			SNP	Ablation	
	Static	Dynamic	External	Static (Small)	Static (Large)	Dynamic	Dynamic	Face-wise	Only Vertices
Error in *mm*	0.23	0.41	0.44	1.67	0.16	0.22	0.14	0.17	0.16
Time in *ms*	7.45	9.87	7.45	7.62	46.61	47.39	46.61	34.92	0.72

4.2.2 Quantitative Analysis. First of all, SoftDECA provides very close approximations in static and dynamic animations with average test reconstruction errors of only $0.22mm$ and $0.41mm$, respectively. Hence, overall, it becomes evident that SoftDECA generalizes across different human identities, facial expressions, and simulation parameters. Nevertheless, the expressions are all obtained from unpersonalized blendshapes which is why we further evaluate on a static external dataset from the 3DScanstore[3]. In this dataset, for each of seven heads, between 20 and 35 scanned facial expressions are available which we convert into personalized ARKit blendshapes using example-based facial rigging [Li et al. 2010]. Starting from there, we create a test dataset as before. Although the 3DScanstore examples are likely not covered by the DECA distribution, the reconstruction error only slightly increases to $0.44mm$.

Despite the high approximation quality, SoftDECA needs only $7.45ms$ to calculate a static frame on average while a runtime of $9.87ms$ is needed for a dynamic frame. This brief runtime makes SoftDECA appealing even for demanding virtual reality applications. For applications in which unseen personalized blendshape are not desired, we also test a variant of SoftDECA that directly predicts vertex positions. This version achieves an accuracy of 0.16 mm and can be accelerated to only $0.71ms$ per frame.

4.2.3 Static Comparisons. For static simulations, SoftDECA can only be compared to SoftSMPL as SNP is solely designed to approximate dynamic effects. Essentially, the difference between the SoftDECA and the SoftSMPL architecture is the difference between our hypernetwork MLP and a standard MLP. SoftSMPL is originally designed for full bodies and has a motion descriptor as input that describes a body and its state. Adapted to our case, these are the blendshape weights, simulation parameters, and the identity code. First, to keep the inference times approximately consistent, we employ the same network dimensions for the standard MLP as in the hypernetwork. As a result, the reconstruction error of the SoftSMPL MLP increases significantly to an average of $1.67mm$. Therefore, we additionally investigate a larger MLP which achieves approximately the same reconstruction error as SoftDECA. In turn, however, the runtime increases tremendously to 46.61ms. Another canonical alternative to the hypernetwork is a standard MLP that in the last layer does not map to all DGs simultaneously but calculates the DGs face-wise. The reconstruction error is low with $0.17mm$, but the runtime is also high with $34.92ms$. Other architectures like CNNs, GNNs, or transformers could not be evaluated in real-time on a consumer-grade CPU with sufficient accuracy.

4.2.4 Dynamic Comparisons. For dynamic simulations, SoftDECA can be compared against both SoftSMPL and SNP. Contrary to SoftDECA, SoftSMPL and SNP compute dynamics in a latent space and not directly on vertices. Both differ from one another in that SoftSMPL additionally relies on a recurrent GRU network [Chung et al. 2014], whereas SNP is purely based on a standard MLP. In both cases, we compare solely with the *larger* network design mentioned earlier since we are mainly interested in evaluating the accuracy of our dynamic approximation and not in comparing runtimes. It can be observed that the SoftSMPL as well as the SNP design achieve slightly improved reconstruction errors with $0.22mm$ and $0.24mm$, respectively. However, since both do not work vertex-wise, they are not suitable for locally varying simulation parameters.

4.2.5 Qualitative Analysis. A visual demonstration of SoftDECA's capabilities is given in Figure 7 where the SoftDECA predictions are contrasted with the targeted physics-based facial simulation. For instance, in a) it can be observed that, although collisions are not guaranteed to be removed, they remain largely dissolved. In b), the triangle strain of the skin is increased locally in the area of the cheeks, leading to the formation of wrinkles in this region. In c), it is demonstrated that external effects can also be included by means of increased gravity. A *surgical manipulation* is shown in d), in which the jaw is lengthened along the vertical axis in the neutral state while the volume of the head is maintained. The representation of a humanoid alien in e) illustrates the robustness of SoftDECA even outside the DECA distribution. This robustness is mainly achieved by transferring DGs instead of directly predicting vertex positions. Our interpretation of zombification is achieved in f) by growing the area of the skin. This effect highlights that SoftDECA is able to closely approximate such excessive high-frequency details, too. Finally, in g-h) we present how different weight additions can be simulated in a non-linear way. For this purpose, we raise the volume of the soft tissue by 20% and 40%. Due to the already comprehensive training domain of SoftDECA, many other effects can be animated in a computationally efficient way that are not displayed in Figure 7. We refer the reader to the supp. material where additional simulations are shown in a video including dynamic effects.

5 LIMITATIONS

Although SoftDECA inherits most of the advantages of physics-based facial animations, it lacks the intrinsic handling of interactive effects such as wind or colliding objects. Moreover, although we allow for extensive localized artistic interventions, mixtures of material properties have not been part of the training data. Incorporating such mixtures into the training data is difficult as it is hard to define an adequate mixture distribution. Nonetheless, the smooth material blending of SoftDECA visually appears to be a sufficient approximation.

6 CONCLUSION

In this work, we presented SoftDECA, a computationally efficient approximation of physics-based facial simulations even on consumer-grade hardware. With a few exceptions, most simulation capabilities are retained, such as dynamic effects, volume preservation, wrinkle generation, and many more. At this, SoftDECA's runtime is attractive for high-performance applications and low-budget hardware. Moreover, it is lightweight to deploy as it generalizes across different head shapes, facial expressions, and material properties. Finally, the ability to make localized changes after training constitutes an attractive framework for artistic customization.

We aim to improve SoftDECA in at least two directions. On the one hand, with an even more accurate anatomical model that represents e.g. trachea and esophagus more precisely. On the other hand, recent results [Romero et al. 2022] show that contact deformations can also be efficiently learned. Since people touch their faces dozens of times [Spille et al. 2021] a day, adding contact-handling for more realistic gestures may improve immersion significantly.

[3]https://www.3dscanstore.com

ACKNOWLEDGMENTS

This research was supported by the German Federal Ministry of Education and Research (BMBF) through the project HiAvA (ID 16SV8785).

REFERENCES

Jascha Achenbach, Robert Brylka, Thomas Gietzen, Katja zum Hebel, Elmar Schömer, Ralf Schulze, Mario Botsch, and Ulrich Schwanecke. 2018. A multilinear model for bidirectional craniofacial reconstruction. In *Proceedings of the Eurographics Workshop on Visual Computing for Biology and Medicine*. 67–76.

Dicko Ali-Hamadi, Tiantian Liu, Benjamin Gilles, Ladislav Kavan, François Faure, Olivier Palombi, and Marie-Paule Cani. 2013. Anatomy transfer. *ACM transactions on graphics (TOG)* 32, 6 (2013), 1–8.

ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. 2022. RigNeRF: Fully Controllable Neural 3D Portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20364–20373.

Michael Bao, Matthew Cong, Stéphane Grabli, and Ronald Fedkiw. 2019. High-quality face capture using anatomical muscles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10802–10811.

Vincent Barrielle, Nicolas Stoiber, and Cédric Cagniart. 2016. Blendforces: A dynamic framework for facial animation. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 341–352.

Thabo Beeler and Derek Bradley. 2014. Rigid stabilization of facial expressions. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–9.

Bernd Bickel, Manuel Lang, Mario Botsch, Miguel A Otaduy, and Markus H Gross. 2008. Pose-Space Animation and Transfer of Facial Details.. In *Symposium on Computer Animation*. 57–66.

Mario Botsch and Leif Kobbelt. 2005. Real-time shape editing using radial basis functions. In *Computer graphics forum*, Vol. 24. Blackwell Publishing, Inc Oxford, UK and Boston, USA, 611–621.

Mario Botsch, Robert Sumner, Mark Pauly, and Markus Gross. 2006. Deformation transfer for detail-preserving surface editing. In *Vision, Modeling & Visualization*. Citeseer, 357–364.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: Fusing constraint projections for fast simulation. *ACM transactions on graphics (TOG)* 33, 4 (2014), 1–11.

Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. 2010. High resolution passive facial performance capture. In *ACM SIGGRAPH 2010 papers*. 1–10.

Christopher Brandt, Elmar Eisemann, and Klaus Hildebrandt. 2018. Hyper-reduced projective dynamics. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, et al. 2022. Authentic volumetric avatars from a phone scan. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–19.

Dan Casas and Miguel A Otaduy. 2018. Learning nonlinear soft-tissue dynamics for interactive avatars. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 1 (2018), 1–15.

Byungkuk Choi, Haekwang Eom, Benjamin Mouscadet, Stephen Cullingford, Kurt Ma, Stefanie Gassel, Suzi Kim, Andrew Moffat, Millicent Maier, Marco Revelant, et al. 2022. Animatomy: an Animator-centric, Anatomically Inspired System for 3D Facial Modeling, Animation and Transfer. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

Matthew Cong and Ronald Fedkiw. 2019. Muscle-based facial retargeting with anatomical constraints. In *ACM SIGGRAPH 2019 Talks*. 1–2.

Matthew Deying Cong. 2016. *Art-directed muscle simulation for high-end facial animation*. Stanford University.

Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. 2021. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics (ToG)* 40, 4 (2021), 1–13.

Stephan J Garbin, Marek Kowalski, Virginia Estellers, Stanislaw Szymanowicz, Shideh Rezaeifar, Jingjing Shen, Matthew Johnson, and Julien Valentin. 2022. VolTeMorph: Realtime, Controllable and Generalisable Animation of Volumetric Representations. *arXiv preprint arXiv:2208.00949* (2022).

Thomas Gietzen, Robert Brylka, Jascha Achenbach, Katja Zum Hebel, Elmar Schömer, Mario Botsch, Ulrich Schwanecke, and Ralf Schulze. 2019. A method for automatic forensic facial reconstruction based on dense statistics of soft tissue thickness. *PloS one* 14, 1 (2019), e0210257.

Benjamin Gilles, Lionel Reveret, and Dinesh K Pai. 2010. Creating and animating subject-specific anatomical models. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 2340–2351.

Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. 2022. Neural head avatars from monocular RGB videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18653–18664.

David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106* (2016).

Daniel Holden, Bang Chi Duong, Sayantan Datta, and Derek Nowrouzezahrai. 2019. Subspace neural physics: Fast data-driven interactive simulation. In *Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 1–12.

Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D avatar creation from hand-held video input. *ACM Transactions on Graphics (ToG)* 34, 4 (2015), 1–14.

Alexandru-Eugen Ichim, Petr Kadleček, Ladislav Kavan, and Mark Pauly. 2017. Phace: Physics-based face modeling and animation. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14.

Alexandru Eugen Ichim, Ladislav Kavan, Merlin Nimier-David, and Mark Pauly. 2016. Building and animating user-specific volumetric face rigs.. In *Symposium on Computer Animation*. 107–117.

Petr Kadleček, Alexandru-Eugen Ichim, Tiantian Liu, Jaroslav Křivánek, and Ladislav Kavan. 2016. Reconstructing personalized anatomical models for physics-based body animation. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–13.

Petr Kadleček and Ladislav Kavan. 2019. Building accurate physics-based face models from data. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–16.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4401–4410.

Marilyn Keller, Silvia Zuffi, Michael J Black, and Sergi Pujades. 2022. OSSO: Obtaining Skeletal Shape from Outside. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20492–20501.

Martin Komaritzan and Mario Botsch. 2018. Projective skinning. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 1 (2018), 1–19.

Martin Komaritzan, Stephan Wenninger, and Mario Botsch. 2021. Inside Humans: Creating a Simple Layered Anatomical Model from Human Surface Scans. *Frontiers in Virtual Reality* 2 (2021), 694244.

Yeara Kozlov, Derek Bradley, Moritz Bächer, Bernhard Thomaszewski, Thabo Beeler, and Markus Gross. 2017. Enriching facial blendshape rigs with physical simulation. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 75–84.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*. PMLR, 3744–3753.

John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. 2014. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)* 1, 8 (2014), 2.

John P Lewis, Jonathan Mooser, Zhigang Deng, and Ulrich Neumann. 2005. Reducing blendshape interference by selected motion attenuation. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*. 25–29.

Hao Li, Thibaut Weise, and Mark Pauly. 2010. Example-based facial rigging. *Acm transactions on graphics (tog)* 29, 4 (2010), 1–6.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

Nadia Maalin, Sophie Mohamed, Robin SS Kramer, Piers L Cornelissen, Daniel Martin, and Martin J Tovée. 2021. Beyond BMI for self-estimates of body size and shape: A new method for developing stimuli correctly calibrated for body composition. *Behavior Research Methods* 53, 3 (2021), 1308–1321.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

Frederic I Parke. 1991. Control parameterization for facial animation. In *Computer Animation'91*. Springer, 3–14.

Cristian Romero, Dan Casas, Maurizio M Chiaramonte, and Miguel A Otaduy. 2022. Contact-centric deformation learning. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–11.

Shunsuke Saito, Zi-Ye Zhou, and Ladislav Kavan. 2015. Computational bodybuilding: Anatomically-based modeling of human bodies. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–12.

Igor Santesteban, Elena Garces, Miguel A Otaduy, and Dan Casas. 2020. SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Humans. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 65–75.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.

Robert Schleicher, Marlies Nitschke, Jana Martschinke, Marc Stamminger, Björn M Eskofier, Jochen Klucken, and Anne D Koelewijn. 2021. BASH: Biomechanical Animated Skinned Human for Visualization of Kinematics and Muscle Activity.. In *VISIGRAPP (1: GRAPP)*. 25–36.

Eftychios Sifakis, Igor Neverov, and Ronald Fedkiw. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. In *ACM*

SIGGRAPH 2005 Papers. 417–425.

Steven L Song, Weiqi Shi, and Michael Reed. 2020. Accurate face rig approximation with deep differential subspace reconstruction. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 34–1.

Jente L Spille, Martin Grunwald, Sven Martin, and Stephanie M Mueller. 2021. Stop touching your face! A systematic review of triggers, characteristics, regulatory functions and neuro-physiology of facial self touch. *Neuroscience & Biobehavioral Reviews* 128 (2021), 102–116.

Sangeetha Grama Srinivasan, Qisi Wang, Junior Rojas, Gergely Klár, Ladislav Kavan, and Eftychios Sifakis. 2021. Learning active quasistatic physics-based models from data. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.

Robert W Sumner and Jovan Popović. 2004. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)* 23, 3 (2004), 399–405.

Lingchen Yang, Byungsoo Kim, Gaspard Zoss, Baran Gözcü, Markus Gross, and Barbara Solenthaler. 2022. Implicit neural representation for physics-driven actuated soft bodies. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–10.

Li Zhang, Noah Snavely, Brian Curless, and Steven M Seitz. 2008. Spacetime faces: High-resolution capture for¨ modeling and animation. In *Data-Driven 3D Facial Animation*. Springer, 248–276.

Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. 2022. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13545–13555.

Wojciech Zielonka, Timo Bolkart, and Justus Thies. 2023. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4574–4584.