

Contact-Centric Deformation Learning

CRISTIAN ROMERO, Universidad Rey Juan Carlos, Spain

DAN CASAS, Universidad Rey Juan Carlos, Spain

MAURIZIO M. CHIARAMONTE, Meta Reality Labs Research, USA

MIGUEL A. OTADUY, Universidad Rey Juan Carlos, Spain

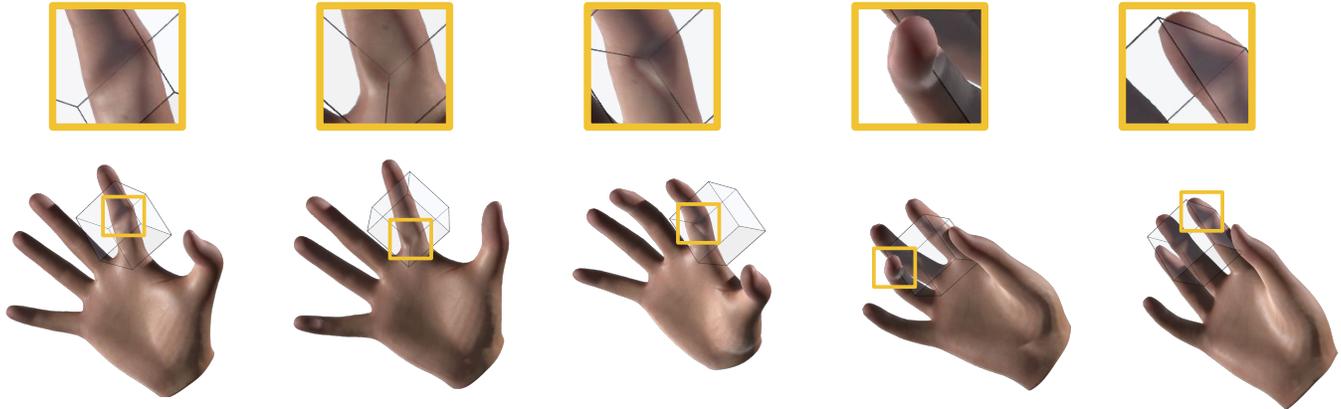


Fig. 1. We present a learning-based method to augment a subspace deformable simulation with contact-driven deformation detail. We learn contact deformations in a contact-centric manner, which allows us to significantly reduce the sampling of configurations of the deformable object, and subsequently learn highly complex deformations. For this real-time simulation of the MANO model [Romero et al. 2017] with dynamics, we used just one pose of the hand for training. Notice the accurate high-resolution deformations due to contact with a rigid object, highlighted in the zoom-ins.

We propose a novel method to machine-learn highly detailed, nonlinear contact deformations for real-time dynamic simulation. We depart from previous deformation-learning strategies, and model contact deformations in a contact-centric manner. This strategy shows excellent generalization with respect to the object’s configuration space, and it allows for simple and accurate learning. We complement the contact-centric learning strategy with two additional key ingredients: learning a continuous vector field of contact deformations, instead of a discrete approximation; and sparsifying the mapping between the contact configuration and contact deformations. These two ingredients further contribute to the accuracy, efficiency, and generalization of the method. We integrate our learning-based contact deformation model with subspace dynamics, showing real-time dynamic simulations with fine contact deformation detail.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: Contact, learning

ACM Reference Format:

Cristian Romero, Dan Casas, Maurizio M. Chiaramonte, and Miguel A. Otaduy. 2022. Contact-Centric Deformation Learning. *ACM Trans. Graph.* 41, 4, Article 70 (July 2022), 11 pages. <https://doi.org/10.1145/3528223.3530182>

Authors’ addresses: Cristian Romero, Universidad Rey Juan Carlos, Madrid, Spain, crisrom002@gmail.com; Dan Casas, Universidad Rey Juan Carlos, Madrid, Spain, dan.casas@urjc.es; Maurizio M. Chiaramonte, Meta Reality Labs Research, Redmond, USA, mchiaram@fb.com; Miguel A. Otaduy, Universidad Rey Juan Carlos, Madrid, Spain, miguel.otaduy@urjc.es.

© 2022 Association for Computing Machinery.

This is the author’s version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, <https://doi.org/10.1145/3528223.3530182>.

1 INTRODUCTION

The simulation of contact and deformations has drawn great interest in computer graphics, as it serves to bring to life computer-generated models of humans and their surrounding objects [McAdams et al. 2011; Smith et al. 2018; Terzopoulos et al. 1987]. However, one of the remaining challenges in the field is to simulate high-resolution contact at interactive rates, e.g., for virtual reality applications.

In our work, we look at leveraging machine-learning methodologies to model contact-driven deformations, inspired by their success in modeling *self-driven deformations* [Pons-Moll et al. 2015; Santesteban et al. 2019; Song et al. 2020], i.e., deformations that emerge as a function of the object’s own motion. These methods employ a subspace representation of the deformable object, and then learn rich nonlinear deformations as a function of the subspace state. Some works have already attempted to model contact deformations using machine-learning approaches, but they either model only smooth global contact response [Holden et al. 2019], or show very limited 3D interactions [Romero et al. 2021].

We hypothesize that there is a fundamental limitation in previous deformation learning strategies. Deformations are modeled in an object-centric manner, which is an excellent choice for self-driven deformations, as they are smooth with respect to the object’s subspace state, and then machine learning achieves good generalization even from sparse data. However, contact-driven deformations are not smooth with respect to the object’s state; therefore, machine-learning these deformations would require dense sampling of the

object’s subspace state. This is hard, as the configuration space may be large and difficult to cover.

We depart from previous deformation-learning strategies, and propose a contact-centric strategy to learn contact-driven deformations. This is also the intuition behind sculpting brushes in digital sculpting [Ferley et al. 1999], and similar to the learning of skeletal deformations in local body-part coordinates [Deng et al. 2020]. We demonstrate that our contact-centric approach shows excellent generalization with respect to the object’s subspace state (down to just 8 training poses for a challenging ‘duck’ example with an 87-dimensional subspace state, or 1 pose for the ‘hand’ example in the teaser figure). Our novel method, presented in Section 3, gathers three main components:

(1) As outlined above, we model contact deformations in a contact-centric manner, i.e., on a local reference of the collider. We observe that contact deformations are smoother when modeled in a contact-centric manner, and this contributes to better generalization, and easier and more accurate learning.

(2) We regard contact deformations as a continuous vector field. Instead of learning a discrete approximation, we learn the continuous field directly, inspired by recent work on implicit surface modeling [Xie et al. 2021]. Learning the contact deformation field generalizes continuity and differentiability to unseen configurations.

(3) We sparsify the mapping between the contact configuration and the resulting contact deformations. In this way, we leverage the locality of contact deformations, and we learn them effectively from sparse data.

In Section 4, we describe the neural-network approximation of our contact deformation model. We also discuss the efficient generation of training data. In Section 5, we discuss the simulation of dynamic deformations using our learning-based contact deformation model. We augment a dynamic subspace deformation with quasi-static contact-driven detail that is expressed in the same subspace, allowing simulations that are both fast and highly detailed.

We have applied our method to real-time dynamic simulations of different deformable objects. We show 2D and 3D subspace simulations generated with the bounded generalized biharmonic coordinates [Wang et al. 2015], and 3D simulations of the MANO hand model [Romero et al. 2017]. We have augmented these dynamic subspace deformations with rich and highly-detailed contact deformations, all in real time. Examples and code are available at <http://mslab.es/projects/ContactCentricLearning/> to aid in the reproduction of our work.

2 RELATED WORK

2.1 Learning-Based Deformation

Nowadays the classic approach to implement deformable 3D shapes is through linear blend skinning (LBS) [Jacobson et al. 2014; Magnat-Thalmann et al. 1988], where an underlying skeleton is used to parameterize the pose of an articulated object, and linear blending of individual bone transformations deforms the shape surface. On top of this, it is common to use a pose-space deformation (PSD) [Lewis et al. 2000] method to mitigate well-known LBS artifacts. PSD adds pose-dependant correctives to the template mesh such that the posed mesh does not exhibit unnatural deformations.

Many works have been proposed to extend PSD in multiple ways. Among those, closest to our work are the methods that *learn* the pose-dependant correctives from data. For full-body humans, SMPL [Loper et al. 2015] learns pose and shape correctives from a large dataset of 4D static scans. Subsequent works have leveraged the learning capabilities of neural networks to extend SMPL to model soft-tissue dynamic deformations [Casas and Otaduy 2018; Pons-Moll et al. 2015; Santesteban et al. 2020]. Similarly, Bailey *et al.* [2018] use multiple neural networks to approximate the rig’s nonlinear deformation components. Some works have also learned correctives for modeling specific body parts, such as faces [Song et al. 2020] or hands [Romero et al. 2017]. Beyond human bodies, garment deformations also have been learned from simulations, using shape correctives [Ma et al. 2020; Patel et al. 2020; Pons-Moll et al. 2017; Santesteban et al. 2019] or even neural features [Zhang et al. 2021].

Despite the realism of the deformations showcased by these methods, due to their self-driven deformation strategy (*i.e.*, deformations depend only on skeletal pose or motion), they are unable to model contact. Our work also stems from the idea of adding learned corrections, but we bring new ingredients to model deformations due to external interactions: a contact-centric representation and a continuous deformation field.

2.2 Subspace and Contact Simulation

We add learning-based contact deformations to subspace dynamic simulations; therefore, we look at how others have designed subspace simulations for deformable objects and how they handled contact.

Model order reduction assumes that a high-resolution deformable object is given, and it finds a low-dimensional subspace that represents accurately the range of deformations of the object [Sifakis and Barbic 2012]. Modal analysis finds a good subspace based on the mechanical properties of the object [Pentland and Williams 1989], and principal component analysis does it based on deformation examples [Krysl et al. 2001]. Modal derivatives can improve the basic linear subspace of these two approaches [Barbič and James 2005], while autoencoders can find a latent nonlinear subspace from data [Fulton et al. 2019]. Recently, Shen et al. [2021] improved the differentiability of deep autoencoders for their use in deformable simulation, and Lee and Carlberg [2021] showed how to enforce physical conservation laws in the learned subspaces. Other model-order reduction approaches can work with artist-driven definitions of subspaces, such as sparse frames [Brandt et al. 2018; Gilles et al. 2011], sparse handles [Wang et al. 2015], or animation rigs [Hahn et al. 2012]. The recent approach of Lan et al. [2020] uses the object’s medial axis to find an expressive geometry-motivated subspace. Finally, some authors have looked at complementing artist-defined subspaces with model-order reduction to augment them with fast dynamics. Some examples include pose-based subspaces [Hahn et al. 2014; Xu and Barbič 2016], local skinned deformations [Tapia et al. 2021], or domain decomposition [Barbič and Zhao 2011; Kim and James 2011; Wu et al. 2015].

A common issue with model order reduction is that contact deformations are not resolved in high detail. The variety of contact

deformations is too large to be captured by the subspace basis, and the resulting simulations appear overly smooth. Some works have addressed this limitation, enriching the subspace with a local basis [Harmon and Zorin 2013] or with a local submesh [Teng et al. 2015]. This last work supports more general deformations than ours, but it suffers a high performance drop as the contact influence grows (they report frame rates below 1 fps). Our method maintains a stable high frame rate (tens of fps), making it suitable for real-time interaction. Romero et al. [2021] combine learning with model order reduction to produce contact-driven deformations. Their method succeeds to produce detailed deformations, but it requires dense sampling of the subspace of the deformable object. Our contact-centric method overcomes this limitation and generalizes well under extremely sparse sampling of the subspace, making it suitable for more complex 3D interactions. In Section 6 we discuss comparisons to the method of Romero et al., which fails to generalize under sparse data. Holden et al. [2019] use machine learning to model the dynamic update of subspace deformable objects under external contact. However, their approach cannot produce detailed contact deformations, as the deformable object’s representation is limited to a linear subspace learned from deformation examples.

2.3 Learning-Based Fields

Our contact-centric approach takes inspiration from the recent trend of learning implicit representations to encode 3D shapes. Initial works learn to approximate the surface of 3D meshes by predicting a binary occupancy of arbitrary 3D points [Chen and Zhang 2019; Mescheder et al. 2019]. Since fully-connected neural networks are used, the learned representation is continuous, memory-efficient, and easily differentiable, which brings many benefits in simulation, computer vision, and geometry processing frameworks. For example, these representations enable differentiable inside/outside queries, which are tricky to implement with traditional representations such as polygonal meshes. Follow-up research [Atzmon and Lipman 2020; Chen and Zhang 2019; Park et al. 2019] demonstrated that neural networks are also capable of learning *distance* to surface, which is also a fundamental building block for many methods in Computer Graphics. Such learning-based encoding, often referred to as *implicit neural representations* or *neural distance fields* [Chibane et al. 2020], has the key benefit of not being bounded by an explicit discretization of the shape surface, which is a fundamental feature for our method.

To train these representations, existing methods often require direct 3D supervision in form of a known or pre-computed implicit representation of the target shape [Chen and Zhang 2019; Park et al. 2019]. Interestingly, more recent methods are able to train directly from raw point clouds (*i.e.*, without supervision at the zero level set) [Atzmon and Lipman 2020; Gropp et al. 2020] or open surfaces [Chibane et al. 2020], something that is not possible with traditional representations of signed distance fields (SDF).

Beyond rigid surfaces, the advantages that learned implicit representations bring have been leveraged to model more complex objects, such as articulated shapes. Deng et al. [2020] model an articulated human body using a piecewise implicit representation. Subsequent works learn fully-parametric body models [Alldieck et al. 2021; Mihajlovic et al. 2021], hands [Karunratanakul et al.

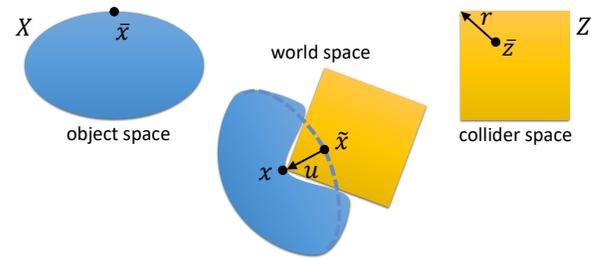


Fig. 2. When a collider Z touches a deformable object X , it produces a displacement field $u(\bar{x})$. We model the full deformation field $x(\bar{x})$ as the sum of a dynamic subspace deformation $\tilde{x}(\bar{x})$ and a learning-based approximation of the contact displacement field $u(\bar{x})$. A key insight of our method is to learn this field as a displacement $r(\bar{z})$ parameterized in collider space.

[2021], hand-and-object interactions [Karunratanakul et al. 2020], garments [Corona et al. 2021], and personalized dressed humans [Saito et al. 2021; Tiwari et al. 2021].

Other methods explore more general uses of these learned fields and, beyond using them to approximate implicit functions, leverage them to expand surface properties to 3D points. This is inspired by non-learning works of Kim et al. [2017] and Romero et al. [2020], who diffuse surface skinning weights inwards to articulate volumetric human bodies represented with tetrahedra. Subsequent learning-based methods expand surface properties, such as skinning weights, *outside* the body surface. This strategy has been used for registering 3D scans to meshes [Bhatnagar et al. 2020], and for articulating raw scans of dressed humans [Huang et al. 2020]. Santesteban et al. [2021] go one step further and learn neural fields to diffuse pose-and-shape surface correctives to \mathbb{R}^3 . Learned fields are used for projecting garments to a canonical shape, which enables highly-efficient handling of body-garment collisions. We also use learned fields to model collisions, but our contact-centric formulation is more general, enables the use of external colliders, and can be plugged into dynamic subspace deformation frameworks.

The high potential of learning field representations has only recently been identified [Xie et al. 2021], and it has quickly extended to address many different problems in Computer Vision and Computer Graphics [Chen et al. 2021]. One prominent example is Neural Radiance Fields (NeRF) [Mildenhall et al. 2020], which learn to synthesize novel views of complex scenes by optimizing a continuous volumetric scene function using a sparse set of input views. Another popular example is methods that learn to reconstruct 3D shapes from images by conditioning an implicit representation on local features extracted from images [Saito et al. 2019, 2020]. In our work, we also leverage this high potential, and we apply it to contact displacement fields.

3 CONTACT-CENTRIC DEFORMATIONS

In this section, we describe how we model contact-driven deformations. Our modeling approach, *i.e.*, the selection of input and output representations of contact-driven deformations, is key for designing an effective learning-based approximation.

We start the section with a definition of the notation, as well as a description of our subspace deformable objects. Then, we define collider-space displacement fields, as a smooth representation of the deformation fields produced by contact. We continue with a discussion of continuous vs. discrete representations of the displacement field, and the impact on the design of a learning-based approximation. To conclude, we propose a sparse approximation of the displacement field to further improve the learning ability.

3.1 Definitions

We learn contact-centric deformations on a subspace deformable object X . In the absence of contact, a point in (undeformed) object space $\bar{x} \in X$ is mapped to a deformed position \tilde{x} in world space through a subspace deformation. In our work, we show different subspace deformation models that combine a dynamic subspace deformation with quasi-static learning-based corrections (also parameterized in the same subspace). More specifically, one example we show is the use of combined point and frame handles, with a smooth deformation field defined by bounded generalized biharmonic coordinates (BGBC) [Wang et al. 2015], and further augmented with learning-based internal corrections [Romero et al. 2021]. Another example that we show is the use of dynamic articulated skeletons, with linear blend skinning, and parametric pose-based corrections [Romero et al. 2017]. We denote as \mathbf{x} the subspace kinematic configuration of the deformable object X .

Let us also consider a collider object Z , and $\bar{z} \in Z$ a point in the collider space. In our work, we limit ourselves to rigid colliders. Then, we denote as \mathbf{z} the rigid configuration of the collider Z .

When the deformable object X touches a collider Z , we augment the subspace deformation field $\tilde{x}(\bar{x})$ with a contact displacement field $u(\bar{x})$, which yields a total deformation field

$$\mathbf{x}(\bar{x}) = \tilde{x}(\bar{x}) + u(\bar{x}), \quad (1)$$

as shown in Figure 2. In our work, we model the subspace deformation $\tilde{x}(\bar{x})$ using dynamics, and the contact displacement $u(\bar{x})$ as a quasi-static deformation. In this way, deformable objects exhibit rich global dynamics combined with contact-driven detail.

3.2 Collider-Space Displacement

We wish to find a suitable parameterization of the contact displacement field $u(\bar{x})$ that allows efficient and accurate approximation with a learning-based architecture. In the limit case of a translation of a collider along a flat, infinite, homogeneous object, the displacement field induced by contact is constant when expressed in *collider space*. In more general cases, the collider may produce a global deformation on the deformable object, but far from the collider this deformation is well captured by the subspace deformation $\tilde{x}(\bar{x})$; it is close to the collider where the additional displacement $u(\bar{x})$ is relevant. We observe that, when the collider moves, this local contact-driven displacement varies more smoothly in collider space than in object space, as shown in Figure 3.

Based on this intuition, we choose to parameterize the contact displacement in collider space, $r(\bar{z})$, as depicted in Figure 2. Then, to evaluate the world-space displacement, we first transform the subspace deformation $\tilde{x}(\bar{x})$ to collider space, and then transform the collider-space displacement again to world space. With $\mathbf{T}(\mathbf{z})$

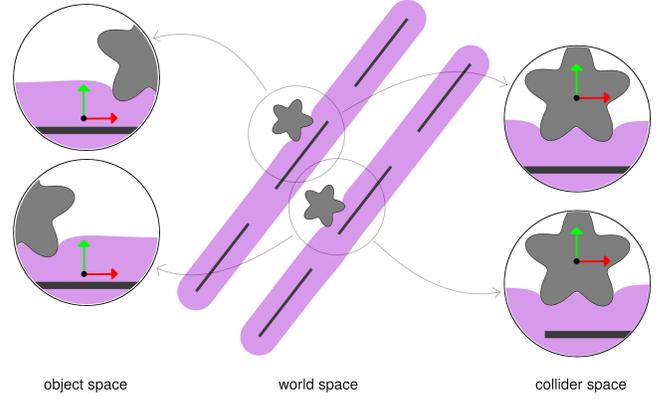


Fig. 3. The close-ups compare the representation of contact displacements in object space \bar{x} (left) vs. collider-space \bar{z} (right) for these two examples. As the collider sweeps through the surface of the deformable object, collider-space contact displacements are notably smoother, and this drastically impacts the learning ability of our method.

a rigid transformation based on the collider's configuration, the displacement is formally obtained as

$$u(\bar{x}) = \mathbf{T}(\mathbf{z}) \cdot r(\bar{z}), \quad \text{with } \bar{z} = \mathbf{T}(\mathbf{z})^{-1} \cdot \bar{x}(\bar{x}). \quad (2)$$

The contact displacement field depends on the relative configuration between the deformable object and the collider, which we express as $\mathbf{T}(\mathbf{z})^{-1} \cdot \mathbf{x}$. In practice, we implement this by transforming all point and frame handles of the deformable object to the local reference frame of the collider. Based on this relative configuration, the contact displacement field can be defined by the following function:

$$r(\bar{z}) \equiv f(\bar{z}, \mathbf{T}(\mathbf{z})^{-1} \cdot \mathbf{x}). \quad (3)$$

We approximate the function f using machine learning. As discussed above, f is in practice a smooth function of the relative configuration between the collider and the deformable object. Typical data-driven deformation methods [Pons-Moll et al. 2015; Santesteban et al. 2019; Song et al. 2020] learn instead object-space deformations $\mathbf{T}(\mathbf{x})^{-1} u(\bar{x})$. However, we have found that the contact displacement parameterized in object space is far less smooth. As a direct consequence, the collider-space displacement function f can be learned using far fewer training data and with a smaller network than an object-space displacement function.

3.3 Learning of a Contact Displacement Field

For dynamic simulation of contact mechanics, the deformation field $\mathbf{x}(\bar{x})$, and therefore the contact displacement $u(\bar{x})$, must be evaluated at two types of points. One type is points on the surface of the deformable object X , for the computation of contact potentials or contact constraints. The other type is points within X , for the computation of the internal energy (and its derivatives). Due to the subspace deformation, in practice we use cubature points in the second case [An et al. 2008].

Both types of evaluation points are fixed in object space \bar{x} . Therefore, under an object-space parameterization of contact displacements, it turns out convenient to learn directly the discrete representation of this function. Furthermore, a common approach in machine learning is to project such high-dimensional representations to a compact linear subspace using PCA, and learn only a small number of PCA coefficients.

However, the evaluation points are not fixed in collider space. While it might be possible to sample the collider space, and apply PCA-based learning, the resulting collider-space displacements should be interpolated to the evaluation points. Instead, motivated by recent methods that learn continuous fields [Xie et al. 2021], we opt to learn the contact displacement function f directly as a vector field.

Furthermore, by learning the field f using a multilayer perceptron (MLP) network, the result is memory-efficient, continuous, and fully differentiable, which are key properties for successful dynamic simulation. The displacement $u(\bar{x})$ is defined only inside the deformable object X , and this leads to a discontinuity in the sampling of the collider space \bar{z} when learning the function f . However, the inductive bias of the MLP network smoothly generalizes to unseen points in the collider space \bar{z} , which may be queried at runtime. The computation of forces and their derivatives requires the evaluation of gradients with respect to the collider space. However, the differentiability of the network provides gradient evaluation by construction.

3.4 Sparsification of the Learning Function

The major challenge in learning the contact displacement function f is the dimensionality of the configuration \mathbf{x} of the deformable object. At first sight, applying our contact-centric deformations to objects with a rich underlying subspace \mathbf{x} (i.e., with many subspace degrees of freedom) requires a combinatorial explosion of the deformed configurations that must be fed as training data to learn f , and a function that is more complex and more challenging to learn.

However, as discussed earlier, we can safely assume that contact displacements have local support, as deformations far from the collider are coarser and well represented by the underlying subspace deformation. Then, the contact displacement at an object-space location \bar{x} is only influenced by the configuration \mathbf{x} of nearby handles of the subspace deformation. Note that, even though the contact displacement function f is parameterized in collider space \bar{z} , it implicitly depends on the object space \bar{x} through $\bar{z} = \mathbf{T}(\mathbf{z})^{-1} \cdot \bar{x}$.

Based on the observations above, we approximate the contact displacement field (3) through a sparse function:

$$r(\bar{z}) \approx f(\bar{z}, \mathbf{W}(\bar{x}) \cdot \mathbf{T}(\mathbf{z})^{-1} \cdot \mathbf{x}). \quad (4)$$

where $\mathbf{W}(\bar{x})$ is a matrix of spatially varying sparsifying weights, i.e., many of its rows are zero.

We leverage the sparsity of our subspace deformation models to define the sparsifying weights. Specifically, with $\mathbf{U}(\bar{x})$ the subspace basis (e.g., BGBC basis or skinning weights) at a material point \bar{x} , we build the weights as $\mathbf{W}(\bar{x}) = \text{diag}(\mathbf{U}(\bar{x}))$. Similar ideas of spatially varying sparse weights have been used in other contexts to obtain local pose definitions, e.g., weighted pose-space deformation [Kurihara and Miyata 2004] or pose attention weights [Saito et al. 2021]

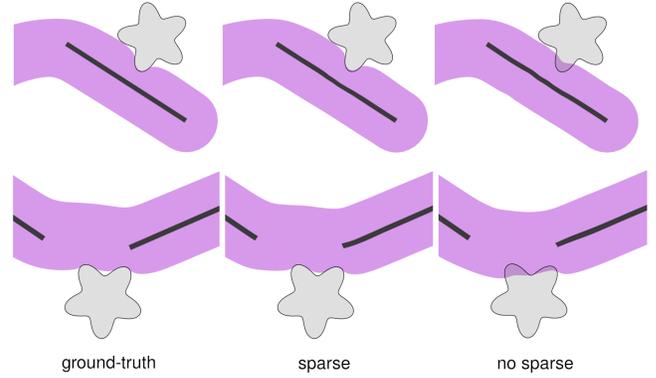


Fig. 4. Two examples (top, bottom) to depict that contact displacements are dominated by the configuration of nearby handles/bones of the deformable object. We leverage this observation designing a sparse approximation of the contact displacement function. Here, we compare ground-truth displacements (left), learned displacements with sparsifying weights, i.e., Eq. (4) (middle), and without sparsifying weights, i.e., Eq. (3) (right). With the same training data, the sparse function achieves superior results, as it succeeds to disambiguate the subspace state that contributes to the contact displacements.

Figure 4 shows an example comparing the learning accuracy with a sparse vs. a dense learning function, and we provide a quantitative analysis in Section 6.

4 DATA AND LEARNING

We pose the problem of designing a learning-based approximation of the contact displacement function f in (4). Solving this problem requires addressing several tasks, which define the structure of this section.

First, we address the design and training of a neural network architecture to compute the contact displacement function f . Second, we describe our strategy for sampling the arguments of f , i.e., the collider space, \bar{z} , and the relative configuration between object and collider, $\mathbf{T}(\mathbf{z})^{-1} \cdot \mathbf{x}$. To conclude, we describe the generation of ground-truth contact displacements $r(\bar{z})$, which requires solving contact configurations with and without contact displacements.

4.1 Neural Network Architecture

We use a fully-connected, 2-layer MLP to model the function f (4), with \tanh as activation function. The actual size of each layer depends on the specific example; see Section 6 for details. Our research focus was on the design of the function to be learned, not on the learning architecture. However, similar to other works that learn fields, we investigated the use of Fourier features [Benbarka et al. 2022; Sitzmann et al. 2020] to improve the learning ability of the neural network. Nevertheless, our initial attempts were not successful, as the generalization outside the sampled region became worse. We leave optimizations of the neural network architecture as future work.

To learn the parameters of the network, we define a loss function that combines two terms. One is the L2 error of estimated contact displacements vs. ground-truth training displacements, summed

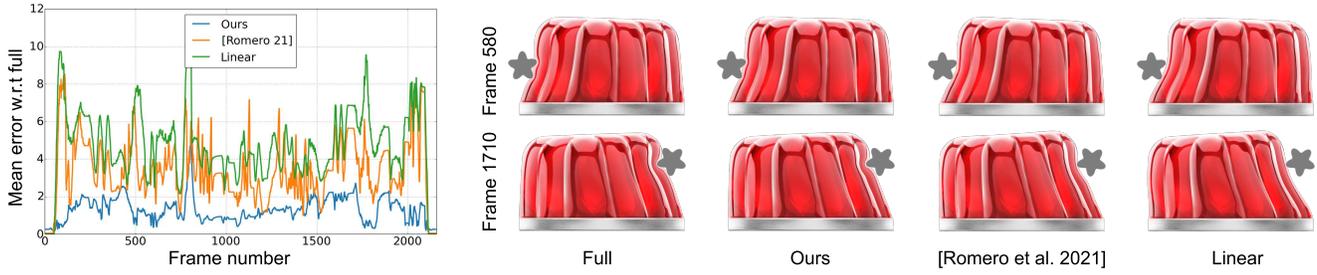


Fig. 5. Our approach significantly improves the generalization capabilities of the state-of-the-art method of Romero *et al* [2021], and closely matches the realism of full simulation. Our method is able to learn the complex interaction between the star-shape collider and the deformable jelly using one order of magnitude less neurons and training data than the original settings used by Romero *et al* [2021]. In contrast, when trained with such reduced dataset, Romero *et al* [2021] are unable to learn deformations due to contact.

over points in the volume of the deformable object X . The other term is the L2 error of differences of estimated contact displacements vs. differences of ground-truth training displacements, summed over surface edges of X . We observed best preservation of contact detail when combining both loss terms.

4.2 Sampling the Contact Displacement Function

The generation of training data requires sampling the arguments of f , i.e., the collider space, \bar{z} , and the relative configuration between object and collider, $\mathbf{T}(\mathbf{z})^{-1} \cdot \mathbf{x}$. Sampling the collider space is easy. Motivated by the loss function defined above, we use the nodes $\{\bar{x}_i\}$ of the volumetric mesh of the deformable object, and we transform them to the collider space through $\mathbf{T}(\mathbf{z})^{-1} \cdot \bar{x}_i$. Note that the collider-space samples vary depending on the relative configuration of the deformable object and the collider.

Sampling the relative configuration between the object and the collider is more challenging. The dimensionality of this space can be seen as the number of degrees of freedom of the deformable object, leaving the collider fixed. However, under this view, the space would be very difficult to sample, as we care only about colliding configurations. Alternatively, we look at the space as the Cartesian product of four subspaces: the configuration \mathcal{X} of the deformable object X after removing rigid transformations, the surface of the deformable object ∂X which defines contact locations, the rotation $SO(3)$ of the collider, and the penetration depth $\mathcal{D} \subset \mathbb{R}$ between collider and deformable object. The full space can be represented as $\mathcal{X} \times \partial X \times SO(3) \times \mathcal{D}$. We sample each of these four subspaces independently. For the penetration depth, we simply use evenly distributed samples up to a maximum depth, with a bit of random noise. For the other three subspaces, we generate a large potential set of samples and we pick a representative subset using a greedy furthest-point strategy (based on surface geodesic distance for ∂X and norm of axis angle for $SO(3)$).

We pay special attention to sampling the rigid-free configuration \mathcal{X} of the deformable object. We start by executing interactive contact simulations between the deformable object and the collider, leveraging the speed of the subspace simulation model of the deformable object. To represent the rigid-free configuration space \mathcal{X} , we build a graph of handle connectivity of the subspace model, and for each

state \mathbf{x} of the deformable object we compute relative handle transformations for all edges in the graph. Given a data set of relative transformations, we normalize separately the entries corresponding to each edge. Based on this definition of rigid-free configurations, for furthest-point selection we use the Euclidean distance between normalized edge transformations.

Thanks to the smoothness of collider-space contact displacements, together with our decomposition of the relative configuration between the object and the collider, and the furthest-point sample selection discussed above, we manage to drastically reduce the number of samples needed in \mathcal{X} , the configuration of the deformable object. This is arguably the hardest subspace to sample, and a naïve learning strategy would require exhaustive exploration of the configuration space. Instead, as shown in our results in Section 6, we sample complex high-dimensional configuration spaces (29 point handles in the ‘duck’) with fewer than 10 configuration samples, yet the learned model generalizes well to unseen states.

4.3 Ground-Truth Contact Displacements

Using the procedure described above, we can sample representative contact configurations of the collider and the deformable object in an efficient manner. Next, for each of these configurations we must compute ground-truth contact displacements $r(\bar{z})$, as the difference between full-space deformations $x(\bar{x})$ and subspace deformations $\tilde{x}(\bar{x})$. However, it is important that the subspace states \mathbf{x} of these deformation fields match.

The subspace deformation is directly given by the interactive generation of contact configurations. Therefore, we are left to compute a full-space deformation constrained to the same subspace state. Even though this task is part of preprocessing, a typical constrained dynamics solve based on Lagrange multipliers could be very time-consuming, due to the large number of simulations and the size of the full-space representation. Instead, we restrict the full-space simulation to the null-space of the subspace using a projection method. Given the basis \mathbf{U} of the subspace, the matrix $\mathbf{P} = \mathbf{I} - \mathbf{U} (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T$ represents a projection to the null-space of the subspace. To run the full-space simulation, we use the modified conjugate gradient method [Ascher and Boxerman 2003], with \mathbf{P} as projection matrix. Note that we do not explicitly compute \mathbf{P} , we only compute the

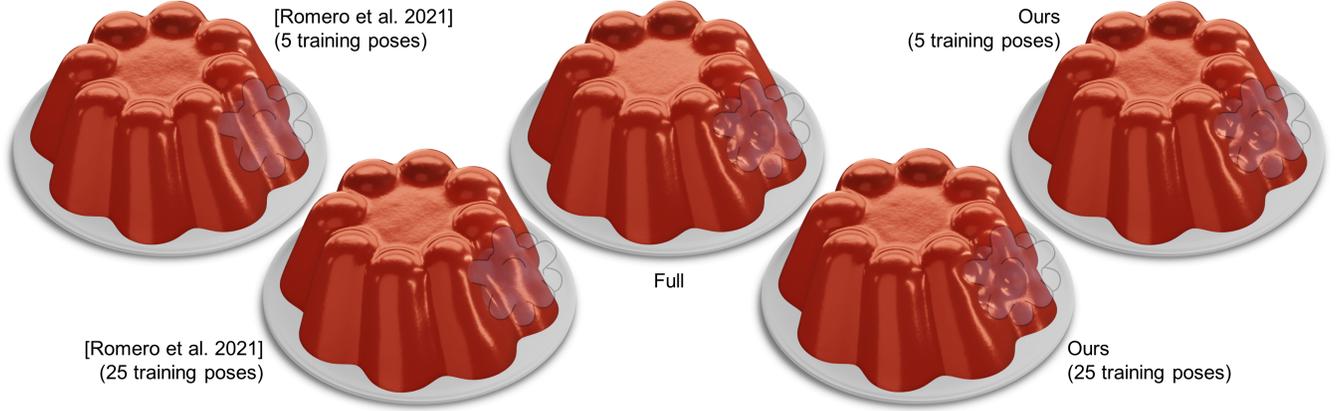


Fig. 6. The generalization capabilities of our collider-centric method are also evident in this 3D jelly example. Our method is accurate when trained with just 5 poses of the jelly, and increasing the number of poses to 25 provides little gain. In contrast, object-centric learning, as done by Romero *et al* [2021], fails to learn contact deformations with 5 poses, and only slightly improves with 25 poses. In Table 2 we provide numerical comparisons. Object-centric learning suffers the curse of dimensionality, and would require an intractable number of training poses.

Cholesky factorization of $\mathbf{U}^T \mathbf{U}$, which is small and fast, and we apply the various matrix multiplications on each iteration of conjugate gradient.

5 SIMULATION OF DYNAMIC DEFORMATIONS

Our novel contact-centric learned deformations can be added to a dynamic subspace simulation model, retaining the fast subspace formulation in the combined simulation. In this section, we formulate the full dynamics problem, paying special attention to the inclusion of the learning-based contact displacement function f in (4).

We find it convenient to formulate dynamics as an optimization problem, using the optimization-formulation of backward Euler [Gast *et al.* 2015; Martin *et al.* 2011]. In this way, we can seamlessly use our deformation field definitions (Section 3.1), integrate quantities on the full-space, and optimize only subspace degrees of freedom. Given a collider configuration \mathbf{z} , and an explicit-Euler update of full-space positions $x^*(\bar{x})$, the subspace configuration of the deformable object is computed as:

$$\mathbf{x} = \arg \min W_{\text{inertial}} + W_{\text{elastic}} + W_{\text{contact}} \quad (5)$$

$$W_{\text{inertial}} = \int_{\mathcal{X}} \frac{\rho}{2h^2} \|x(\bar{x}, \mathbf{x}, \mathbf{z}) - x^*(\bar{x})\|^2 d\bar{x},$$

$$W_{\text{elastic}} = \int_{\mathcal{X}} \Psi(x(\bar{x}, \mathbf{x}, \mathbf{z})) d\bar{x},$$

$$W_{\text{contact}} = \int_{\partial X} \Phi(\mathbf{T}(\mathbf{z})^{-1} \cdot x(\bar{x}, \mathbf{x}, \mathbf{z})) d\bar{x}.$$

Here, ρ is the mass density of the object, h is the time step, and the full-space deformation field $x(\bar{x})$ is defined by combining (1), (2) and (4):

$$x(\bar{x}, \mathbf{x}, \mathbf{z}) = \tilde{x}(\bar{x}, \mathbf{x}) \quad (6)$$

$$+ \mathbf{T}(\mathbf{z}) \cdot f(\bar{\mathbf{z}} = \mathbf{T}(\mathbf{z})^{-1} \cdot \tilde{x}(\bar{x}, \mathbf{x}), \mathbf{W}(\bar{x}) \cdot \mathbf{T}(\mathbf{z})^{-1} \cdot \mathbf{x}).$$

Table 1. Details about dataset size and runtime performance for the different objects used to showcase our method. For descriptions about sample types (e.g., \mathcal{X} , ∂X , $\text{SO}(3)$, \mathcal{D}), see Section 4.2.

| Example | handles point/bone | mesh size tris/tets | # samples | | | | neurons | fps | | |
|----------|-----------------------|------------------------|---------------|--------------|----------------|---------------|---------|--------|------|------|
| | | | \mathcal{X} | ∂X | $\text{SO}(3)$ | \mathcal{D} | | linear | full | ours |
| Duck | 29/0 | 37,049 | 8 | 64 | 30 | 5 | 500 | 277 | 2 | 25 |
| Floater | 24/0 | 39,450 | 8 | 64 | 30 | 5 | 500 | 211 | 1 | 23 |
| Hand | 0/16 | 82,395 | 1 | 300 | 48 | 5 | 500 | 81 | 1 | 36 |
| Jelly 2D | 8/1 | 13,720 | 5 | 21 | 32 | 5 | 300 | 548 | 5 | 53 |
| Jelly 3D | 18/1 | 60,830 | 5 | 64 | 30 | 5 | 500 | 191 | 1 | 19 |
| Worm | 0/3 | 20,213 | 7 | 64 | 32 | 5 | 300 | 428 | 3 | 50 |

In this expression, we show the explicit dependencies of the deformable object configuration \mathbf{x} , as these are important for the evaluation of gradients.

In (5) above, Ψ is an elastic energy model. In our case, we have used the stable Neo-Hookean formulation. Φ is a contact potential based on a signed distance field precomputed for the collider. It is zero for negative distances and cubic for positive distances. We integrate the inertial and elastic terms using cubature [An *et al.* 2008], with cubature points and weights estimated using a data-oblivious approach [Tapia *et al.* 2021]. We integrate the contact term using all surface mesh points instead.

To solve the optimization (5), we use a Newton-CG solver. To evaluate gradients of the learning function $\frac{\partial f}{\partial \mathbf{x}}$, we perform back propagation on the neural network. We do not store the Hessian explicitly, but instead execute Hessian-vector products. In this regard, we ignore the Hessian of the learning function, and we implement gradient-vector products through an auxiliary back propagation step [Romero *et al.* 2021]. Despite the small size of the subspace Hessian, we found more convenient to use Newton-CG than a direct solver, as each Newton step required very few CG iterations in practice, and hence minimized the number of network evaluations.

6 RESULTS

In this section we quantitatively and qualitatively evaluate our method in a variety of objects, scenarios, and interactions. Additionally, we compare it to the state-of-the-art method of Romero *et al.* [2021], who also model contact deformations using a data-driven approach. As baseline, we also show results using a linear subspace model based on Wang *et al.* [2015]. All the examples were executed on an Intel Core i7-7700K 4-core 4.20 GHz PC with 32 GB of RAM.

In Table 1 we provide details of the objects and datasets used to generate our results, including the mesh discretization of the full-space simulations.

Jelly 2D. Figure 5 shows frames of a star-shape collider interacting with a 2D jelly, using 4 different methods: full simulation, our contact-centric approach, the state-of-the-art method of Romero *et al.* [2021], and the linear method of Wang *et al.* [2015]. Note that we used the original dataset and code publicly available from Romero *et al.* [2021] but, to stress the generalization capabilities of our approach, we used a reduced version of the dataset, consisting of 16,800 ground truth samples ($\approx 15\times$ less than their dataset). To quantitatively evaluate our results, we also plot the mean per-vertex error of each method through a test sequence of more than 2,000 frames. Notice, moreover, that our model requires only 300 neurons while Romero *et al.* [2021] require 3,000 neurons.

Jelly 3D. Figure 6 shows a similar comparison in 3D, with a pointy collider interacting with a 3D jelly. We demonstrate that our collider-centric learning approach is accurate when trained with just 5 configurations of the jelly. The training cost is dominated by the simulation of all training samples (48,000 in total, accounting for all sampled configurations of the collider), which took 37 hours. With the same training data, object-centric learning [Romero *et al.* 2021] fails to produce accurate results. Object-centric learning suffers the curse of dimensionality, and multiplying the training data to 25 configurations of the jelly (184 hours) barely improved the results. The errors for all configurations are also compared numerically in Table 2. The test examples have been generated by projecting static full-space deformations to the handle-based subspace, and then adding learning-based contact deformations. Errors are normalized with respect to the difference between full-space and linear subspace deformation.

These results show that, when using the same amount of training data, our method generalizes much better than the state-of-the-art approach by Romero *et al.* [2021], and it closely reproduces the realism of a full simulation. Importantly, our method is not only able to train with less data, but also to model more complex and highly deformable contact interactions at real-time framerates.

Table 2. Relative error for the different methods and training settings of the 3D Jelly example shown in Figure 6.

| # samples \mathcal{X} | 5 | 25 |
|-----------------------------|-----|-----|
| Ours | 57% | 56% |
| [Romero <i>et al.</i> 2021] | 99% | 86% |

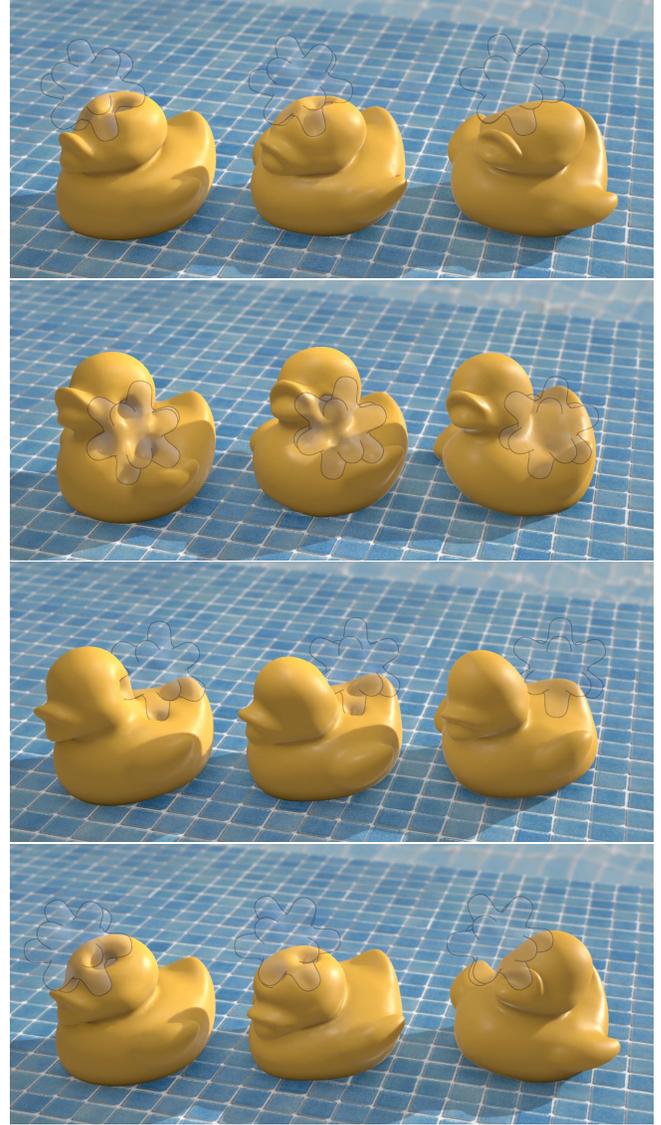


Fig. 7. Qualitative evaluation. We show 4 frames of a sequence where a collider (semitransparent, for better visualization) interacts with a rubber duck. Our method (center), closely matches the natural deformations due to contact that emerge using a full simulation model (left). In contrast, a linear model [Wang *et al.* 2015] (right) is unable to deform correctly.

Worm 2D. We have used the worm in Figures 3 and 4 to evaluate the qualitative and quantitative effect of the sparsification of the

Table 3. Relative error in the Worm example in Figures 3 and 4, with and without sparsification, for different amounts of training data.

| # samples \mathcal{X} | 1 | 3 | 7 |
|-------------------------|-----|-----|-----|
| sparse | 44% | 43% | 39% |
| no sparse | 96% | 76% | 47% |

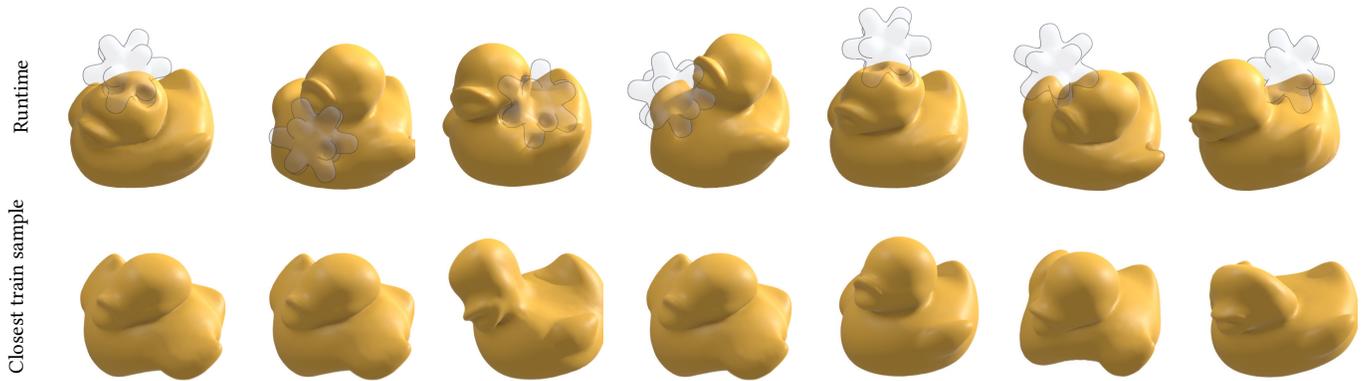


Fig. 8. To depict the good generalization capabilities of our method, here we visualize the *closest* training sample (bottom) to a wide range of different states of the deformable duck (top). Frames were randomly picked from a sequence where the collider interacts in real time with the duck. For this particular demo, we use only 8 samples of the duck state \mathcal{X} to train. Since our approach is collider-centric, it generalizes well to unseen states of the deformable duck.

learning function. We have compared the error with respect to a full-space simulation, with and without sparsification, for different amounts of training data (varying the samples of the worm’s configuration space). The errors reported in Table 3 confirm that sparsification allows a drastic reduction in the amount of training data required.

Hand. Figure 1 and the supplementary video showcase an interactive sequence where a 3D hand manipulates a rigid cube. The subspace deformation is built with the MANO model [Romero et al. 2017], and the skeleton is dynamically simulated. The runtime interaction was produced interactively with a LeapMotion device for hand tracking, and commanding the hand’s skeleton through spring forces. Notice how the skin surface of the hand naturally deforms when it touches the cube, even on sharp edges and corners, all in real time. This example was trained on a single flat pose of the hand. Despite such extremely simple sampling of the configuration space of the hand, our contact-centric formulation, together with the sparsifying weights, achieve excellent generalization to unseen hand poses. We demonstrate that the accuracy of the contact deformation is well-kept across all hand regions, including palm and fingertips, and for any hand pose.

Ducks. Figure 7 shows frames of a sequence where a collider closely interacts with a rubber duck, and qualitatively compares our results to the deformations obtained with a linear subspace model [Wang et al. 2015] and a full-space simulation. The linear model, bounded by the limited expressivity of the subspace, is unable to reproduce the deformations due to contact, producing an unnatural behavior. In contrast, our approach is capable of accurately modeling deformations due to contact, closely matching the realism of the full-space simulation, even in situations with strong interactions and heavily deformed states. Please see the supplementary video for animated results.

In Figure 8, to qualitatively evaluate the generalization capabilities of our approach, we show frames of a sequence where we interactively manipulate the duck with a collider. For each frame, we show the closest duck deformation in the training set. Notice

that, even when the closest sample is far from the current state of the duck, our method is able to accurately reproduce the deformations due to contact. In the particular case of the Duck scenario, our contact-centric representation is able to learn accurate contact with a deformable object using as few as 8 deformed examples. Importantly, even if we learn deformations without dynamics or friction, at runtime our method generalizes well to those settings.

Floater. Our method does not explicitly rely on geometric features of the deformable object or the collider, such as genus or symmetries. Figure 9 shows a scene with a deformable floater of genus 1 and a shell collider. The floater falls on top of the shell and it deforms, to let the shell pass through. With a linear subspace method, the shell fails to pass through the floater, as the necessary deformations are not well represented. Our method, on the other hand, represents them correctly. Please see the motion in the accompanying video.

7 CONCLUSIONS AND FUTURE WORK

In this work, we have presented a contact-centric method to learn contact-driven deformations. These deformations are added to a subspace dynamic simulation model, to produce real-time dynamic simulations of deformable objects with rich contact detail. We have demonstrated that contact-centric parameterization of the learning function drastically simplifies its complexity: the space of configurations can be sparsely sampled, and the resulting learning models are smaller, more efficient, and easier to learn. We further complement contact-centric modeling with a continuous field representation and sparsification of the learning function, which contribute to excellent generalization capabilities.

Our work is not free of limitations, and some of these suggest directions for non-trivial future work. We learn a deformation model per collider object. This could find applicability in interactive applications where colliders are known in advance, but it fails to address applications where multiple colliders interact in complex ways or where colliders are defined dynamically. We also assume that the collider is rigid, which again covers a large set of use cases, but it does not account for deformable-deformable contact.



Fig. 9. A full-space floater (left) falls on top of a rigid shell and it deforms, to let the shell pass through. This motion is well represented with our method (middle), while a linear subspace (right) fails to represent the necessary deformations, and the shell gets stuck.

We have introduced our contact-centric modeling approach in the context of deformable object simulation. However, this approach might find applicability in other problems of interaction with objects, such as joint tracking of hands and objects, or grasp synthesis.

ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers for their helpful comments. We also thank Mickeal Verschoor and Suzanne Sorli for help with the hand demo. This work was funded in part by the European Research Council (ERC Consolidator Grant 772738 *TouchDesign*).

REFERENCES

- Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. 2021. imGHUM: Implicit Generative Models of 3D Human Shape and Articulated Pose. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 5461–5470.
- Steven S. An, Theodore Kim, and Doug L. James. 2008. Optimizing Cubature for Efficient Integration of Subspace Deformations. *ACM Trans. Graph.* 27, 5, Article 165 (2008).
- Uri M. Ascher and Eddy Boxerman. 2003. On the Modified Conjugate Gradient Method in Cloth Simulation. *Vis. Comput.* 19, 7-8 (2003).
- Matan Atzmon and Yaron Lipman. 2020. SAL: Sign Agnostic Learning of Shapes from Raw Data. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*.
- Stephen W. Bailey, Dave Otte, Paul Dilorenzo, and James F. O'Brien. 2018. Fast and Deep Deformation Approximations. *ACM Trans. Graph.* 37, 4 (2018).
- Jernej Barbič and Doug L. James. 2005. Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models. *ACM Trans. Graph.* 24, 3 (July 2005), 982–990.
- Jernej Barbič and Yili Zhao. 2011. Real-Time Large-Deformation Substructuring. In *ACM SIGGRAPH 2011 Papers* (Vancouver, British Columbia, Canada) (*SIGGRAPH '11*). Association for Computing Machinery, New York, NY, USA, Article 91, 8 pages.
- Nuri Benbarka, Timon Höfer, Hamd ul-Moqet Riaz, and Andreas Zell. 2022. Seeing Implicit Neural Representations As Fourier Series. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2041–2050.
- Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. 2020. LoopReg: Self-supervised Learning of Implicit Surface Correspondences, Pose and Shape for 3D Human Mesh Registration. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Christopher Brandt, Elmar Eisemann, and Klaus Hildebrandt. 2018. Hyper-Reduced Projective Dynamics. *ACM Trans. Graph.* 37, 4, Article 80 (2018).
- 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), 10.
- Peter Yichen Chen, Maurizio Chiaramonte, Eitan Grinspun, and Kevin Carlberg. 2021. Model reduction for the material point method via learning the deformation map and its spatial-temporal gradients. *arXiv preprint arXiv:2109.12390* (2021).
- Zhiqin Chen and Hao Zhang. 2019. Learning implicit fields for generative shape modeling. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 5939–5948.
- Julian Chibane, Aymen Mir, and Gerard Pons-Moll. 2020. NASA: Neural Unsigned Distance Fields for Implicit Function Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Enric Corona, Albert Pumarola, Guillem Alenyà, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. SMPLicit: Topology-aware Generative Model for Clothed People. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*.
- Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. 2020. NASA: Neural Articulated Shape Approximation. In *Proc. of European Conference on Computer Vision (ECCV)*.
- Eric Ferley, Marie-Paule Cani, and Jean-Dominique Gascuel. 1999. Practical Volumetric Sculpting. *The Visual Computer* 16 (09 1999).
- Lawson Fulton, Vismay Modi, David Duvenaud, David I. W. Levin, and Alec Jacobson. 2019. Latent-space Dynamics for Reduced Deformable Simulation. *Computer Graphics Forum* 38, 2 (2019), 379–391.
- Theodore F Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M Teran. 2015. Optimization integrator for large time steps. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 21, 10 (2015), 1103–1115.
- Benjamin Gilles, Guillaume Bousquet, Francois Faure, and Dinesh K. Pai. 2011. Frame-Based Elastic Models. *ACM Trans. Graph.* 30, 2 (2011).
- Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. 2020. Implicit Geometric Regularization for Learning Shapes. In *Proceedings of Machine Learning and Systems* 2020. 3569–3579.
- Fabian Hahn, Sebastian Martin, Bernhard Thomaszewski, Robert Sumner, Stelian Coros, and Markus Gross. 2012. Rig-Space Physics. *ACM Trans. Graph.* 31, 4, Article 72 (July 2012), 8 pages.
- Fabian Hahn, Bernhard Thomaszewski, Stelian Coros, Robert W. Sumner, Forrester Cole, Mark Meyer, Tony DeRose, and Markus Gross. 2014. Subspace Clothing Simulation Using Adaptive Bases. *ACM Trans. Graph.* 33, 4 (2014).
- David Harmon and Denis Zorin. 2013. Subspace Integration with Local Deformations. *ACM Trans. Graph.* 32, 4, Article 107 (July 2013), 10 pages.
- 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*.
- Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. 2020. ARCH: Animatable Reconstruction of Clothed Humans. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 3093–3102.
- Alec Jacobson, Zhigang Deng, Ladislav Kavan, and JP Lewis. 2014. Skinning: Real-time Shape Deformation. In *ACM SIGGRAPH 2014 Courses*.
- Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. 2021. A Skeleton-Driven Neural Occupancy Representation for Articulated Hands. In *International Conference on 3D Vision (3DV)*.
- Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. 2020. Grasping Field: Learning Implicit Representations for Human Grasps. In *International Conference on 3D Vision*.
- Meekeyoung Kim, Gerard Pons-Moll, Sergi Pujades, Seungbae Bang, Jinwook Kim, Michael J. Black, and Sung-Hee Lee. 2017. Data-Driven Physics for Human Soft

- Tissue Animation. *ACM Trans. Graph.* 36, 4, Article 54 (2017), 12 pages. <https://doi.org/10.1145/3072959.3073685>
- Theodore Kim and Doug L. James. 2011. Physics-Based Character Skinning Using Multi-Domain Subspace Deformations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 63–72.
- P. Krysl, S. Lall, and J. E. Marsden. 2001. Dimensional model reduction in non-linear finite element dynamics of solids and structures. *Internat. J. Numer. Methods Engrg.* 51, 4 (2001), 479–504.
- Tsuneya Kurihara and Natsuki Miyata. 2004. Modeling Deformable Human Hands from Medical Images. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 355–363.
- Lei Lan, Ran Luo, Marco Fratarcangeli, Weiwei Xu, Huamin Wang, Xiaohu Guo, Junfeng Yao, and Yin Yang. 2020. Medial Elastics: Efficient and Collision-Ready Deformation via Medial Axis Transform. *ACM Trans. Graph.* 39, 3, Article 20 (2020).
- Kookjin Lee and Kevin T. Carlberg. 2021. Deep Conservation: A Latent-Dynamics Model for Exact Satisfaction of Physical Conservation Laws. In *AAAI*.
- J. P. Lewis, Matt Corder, and Nickson Fong. 2000. Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. ACM Press/Addison-Wesley Publishing Co., USA, 165–172.
- Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-person Linear Model. *ACM Trans. Graph.* 34, 6, Article 248 (Oct. 2015), 16 pages.
- Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. 2020. Learning to Dress 3D People in Generative Clothing. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*.
- N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. 1988. Joint-dependent Local Deformations for Hand Animation and Object Grasping. In *Proceedings on Graphics Interface '88* (Edmonton, Alberta, Canada). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 26–33.
- Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-Based Elastic Materials. *ACM Trans. Graph.* 30, 4, Article 72 (2011).
- Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. In *ACM SIGGRAPH 2011 Papers* (Vancouver, British Columbia, Canada) (SIGGRAPH '11). Association for Computing Machinery, New York, NY, USA, Article 37.
- Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*.
- Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. 2021. LEAP: Learning Articulated Occupancy of People. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proc. of European Conference on Computer Vision (ECCV)*.
- Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*.
- Chaitanya Patel, Zhouyincheng Liao, and Gerard Pons-Moll. 2020. TailorNet: Predicting Clothing in 3D as a Function of Human Pose, Shape and Garment Style. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.
- Alex Pentland and John Williams. 1989. Good Vibrations: Modal Dynamics for Graphics and Animation. *Computer Graphics* 23, 3 (1989), 215–222.
- Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. 2017. ClothCap: Seamless 4D Clothing Capture and Retargeting. *ACM Transactions on Graphics (Proc. of SIGGRAPH)* 36, 4, Article 73 (2017), 15 pages.
- Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. 2015. Dyna: A Model of Dynamic Human Shape in Motion. *ACM Trans. Graph.* 34, 4, Article 120 (July 2015), 14 pages.
- Cristian Romero, Dan Casas, Jesús Pérez, and Miguel Otaduy. 2021. Learning Contact Corrections for Handle-Based Subspace Dynamics. *ACM Trans. Graph.* 40, 4 (2021).
- Cristian Romero, Miguel A. Otaduy, Dan Casas, and Jesus Perez. 2020. Modeling and Estimation of Nonlinear Skin Mechanics for Animated Avatars. *Computer Graphics Forum (Proc. Eurographics)* 39, 2 (2020).
- Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Trans. Graph.* 36, 6, Article 245 (2017).
- Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization.
- Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. 2020. PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*.
- Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J. Black. 2021. SCANimate: Weakly Supervised Learning of Skinned Clothed Avatar Networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*.
- Igor Santesteban, Elena Garces, Miguel A. Otaduy, and Dan Casas. 2020. SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Humans. *Computer Graphics Forum* 39, 2 (2020), 65–75.
- Igor Santesteban, Miguel A. Otaduy, and Dan Casas. 2019. Learning-Based Animation of Clothing for Virtual Try-On. *Computer Graphics Forum* 38, 2 (2019), 355–366.
- Igor Santesteban, Nils Thuerey, Miguel A Otaduy, and Dan Casas. 2021. Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021).
- Siyuan Shen, Yin Yang, Tianjia Shao, He Wang, Chenfanfu Jiang, Lei Lan, and Kun Zhou. 2021. High-Order Differentiable Autoencoder for Nonlinear Model Reduction. *ACM Trans. Graph.* 40, 4, Article 68 (2021).
- Eftychios Sifakis and Jernej Barbic. 2012. FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction. In *ACM SIGGRAPH 2012 Courses*. 20:1–20:50.
- Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. 2020. Implicit Neural Representations with Periodic Activation Functions. In *Proc. NeurIPS*.
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (2018).
- Steven L. Song, Weiqi Shi, and Michael Reed. 2020. Accurate Face Rig Approximation with Deep Differential Subspace Reconstruction. *ACM Trans. Graph.* 39, 4 (2020). <https://doi.org/10.1145/3386569.3392491>
- Javier Tapia, Cristian Romero, Jesús Pérez, and Miguel A. Otaduy. 2021. Parametric Skeletons with Reduced Soft-Tissue Deformations. *Computer Graphics Forum* (2021).
- Yun Teng, Mark Meyer, Tony DeRose, and Theodore Kim. 2015. Subspace Condensation: Full Space Adaptivity for Subspace Deformations. *ACM Trans. Graph.* 34, 4, Article 76 (July 2015), 9 pages.
- Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically Deformable Models. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. Association for Computing Machinery, New York, NY, USA, 205–214.
- Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. 2021. Neural-GIF: Neural generalized implicit functions for animating people in clothing. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 11708–11718.
- Yu Wang, Alec Jacobson, Jernej Barbic, and Ladislav Kavan. 2015. Linear Subspace Design for Real-Time Shape Deformation. *ACM Trans. Graph.* 34, 4, Article 57 (2015).
- Xiaofeng Wu, Rajaditya Mukherjee, and Huamin Wang. 2015. A Unified Approach for Subspace Simulation of Deformable Bodies in Multiple Domains. *ACM Trans. Graph.* 34, 6, Article 241 (2015).
- Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. 2021. Neural Fields in Visual Computing and Beyond. [arXiv:arXiv:2111.11426](https://arxiv.org/abs/2111.11426) <https://neuralfields.cs.brown.edu/>
- Hongyi Xu and Jernej Barbic. 2016. Pose-Space Subspace Dynamics. *ACM Trans. Graph.* 35, 4, Article 35 (July 2016), 14 pages.
- Meng Zhang, Tuanfeng Y. Wang, Duygu Ceylan, and Niloy J. Mitra. 2021. Dynamic Neural Garments. *ACM Trans. Graph.* 40, 6, Article 235 (2021).