Tesis Doctoral

Data-driven models of 3D avatars and clothing for virtual try-on

Autor:

Igor Santesteban Garay

Directores:

Dan Casas Guix

Miguel A. Otaduy Tristán

Programa de Doctorado en Tecnologías de la Información y las Comunicaciones

Escuela Internacional de Doctorado

2022

Abstract

Clothing plays a fundamental role in our everyday lives. When we choose clothing to buy or wear, we guide our decisions based on a combination of fit and style. For this reason, the majority of clothing is purchased at brick-and-mortar retail stores, after physical try-on to test the fit and style of several garments on our own bodies. Computer graphics technology promises an opportunity to support online shopping through virtual try-on, but to date virtual try-on solutions lack the responsiveness of a physical try-on experience. This thesis works towards developing new virtual try-on solutions that meet the demanding requirements of accuracy, interactivity and scalability. To this end, we propose novel data-driven models for 3D avatars and clothing that produce highly realistic results at a fraction of the computational cost of physics-based approaches. Throughout the thesis we also address common limitations of data-driven methods by using self-supervision mechanisms to enforce physical constraints and reduce the dependency on ground-truth data. This allows us to build efficient and accurate models with minimal preprocessing times.

Acknowledgements

This thesis is the culmination of several years of hard work in which I have had the privilege of working with many wonderful people. First of all, I would like to thank Dan Casas and Miguel A. Otaduy for giving me the opportunity to embark on this journey with them, and Alex García-Alonso for encouraging me to follow this path in the first place. Nils Thuerey, for all his contributions and invaluable support over the years. Elena Garcés, for all the things I have learned working with her. Maurizio Chiaramonte and Joey Greer, for being a joy to work with and helping me appreciate my work and myself a little more. Many thanks also to all my colleagues and friends, who have been a fundamental support during these years.

I would not have made it this far in life without the public education system of my country, so I would like to take this opportunity to thank the Spanish society as a whole, whose taxes make all this possible. I am also very grateful for my family and all the patience they have had with me over the years.

Thank you!

Contents

Abstract iii

1 Introduction 1
- 1.1 Virtual try-on . . . 2
- 1.2 Open problems . . . 3
- 1.3 Contributions and publications . . . 5

2 Background 7
- 2.1 Virtual avatars . . . 7
  - 2.1.1 Parametric human models . . . 7
  - 2.1.2 Soft-tissue deformation . . . 8
- 2.2 Virtual garments . . . 10
  - 2.2.1 Design and modeling . . . 10
  - 2.2.2 Cloth deformation . . . 11

3 Supervised learning of soft-tissue deformations 15
- 3.1 Introduction . . . 15
- 3.2 Method . . . 17
  - 3.2.1 Human model . . . 18
  - 3.2.2 Soft-tissue deformations subspace . . . 18
  - 3.2.3 Soft-tissue deformation regressor . . . 20
- 3.3 Disentangled motion descriptor . . . 21
  - 3.3.1 Static pose disentanglement . . . 21
  - 3.3.2 Avoiding dynamic pose entanglement . . . 23
- 3.4 Implementation details . . . 24
  - 3.4.1 Soft-tissue autoencoder and regressor . . . 24
  - 3.4.2 Pose autoencoder . . . 25
- 3.5 Evaluation . . . 25
  - 3.5.1 Soft-tissue autoencoder evaluation . . . 25
  - 3.5.2 Soft-tissue regressor evaluation . . . 26
  - 3.5.3 Runtime performance . . . 29
- 3.6 Conclusions . . . 29

4 Supervised learning of garment deformations **31**
4.1 Introduction . . . 31
4.2 Method . . . 33
4.2.1 Clothing model . . . 34
4.2.2 Garment fit regressor . . . 36
4.2.3 Garment wrinkle regressor . . . 37
4.3 Implementation details . . . 38
4.3.1 Dataset . . . 39
4.3.2 Networks and training . . . 40
4.4 Evaluation . . . 40
4.4.1 Quantitative evaluation . . . 41
4.4.2 Qualitative evaluation . . . 42
4.4.3 Runtime performance . . . 45
4.5 Conclusions . . . 46

5 Self-supervised learning of garment deformations **47**
5.1 Introduction . . . 47
5.2 Method . . . 49
5.2.1 Garment model . . . 50
5.2.2 Optimization-based dynamic deformation . . . 51
5.2.3 Turning dynamics into self-supervision . . . 51
5.2.4 Material model . . . 52
5.2.5 Regressing garment deformations . . . 54
5.3 Evaluation . . . 55
5.3.1 Training . . . 55
5.3.2 Quantitative evaluation . . . 56
5.3.3 Qualitative evaluation . . . 58
5.4 Conclusions . . . 60

6 Handling collisions between garments and bodies **61**
6.1 Introduction . . . 61
6.2 Overview . . . 63
6.3 Canonical space of garment deformations . . . 64
6.3.1 Diffused human model . . . 65
6.3.2 Garment model . . . 66
6.3.3 Projecting the ground-truth data . . . 67
6.3.4 Generative garment deformation subspace . . . 69
6.4 Regressing garment deformations . . . 71
6.5 Implementation details . . . 72

6.5.1 Dataset . . . 72
6.5.2 Neural networks . . . 72
6.6 Evaluation . . . 74
6.6.1 Quantitative evaluation . . . 74
6.6.2 Qualitative evaluation . . . 75
6.6.3 Runtime performance . . . 77
6.7 Conclusions . . . 78

7 Handling collisions between layered garments 79
7.1 Introduction . . . 79
7.2 Neural fields . . . 81
7.3 Untangled layered neural fields . . . 82
7.3.1 Implicit surface model . . . 82
7.3.2 Neural untangling . . . 84
7.4 Mix-and-match virtual try-on . . . 85
7.4.1 Explicit garment model . . . 86
7.4.2 Optimization of untangled garments . . . 87
7.5 Implementation details . . . 88
7.5.1 Per-garment preprocess . . . 88
7.5.2 Untangling operator. . . . 90
7.5.3 Optimization. . . . 91
7.6 Evaluation . . . 91
7.6.1 Quantitative evaluation . . . 91
7.6.2 Qualitative evaluation . . . 92
7.7 Conclusions . . . 94

8 Conclusions 95
8.1 General discussion . . . 95
8.2 Final remarks . . . 97

Bibliography 99

A Resumen 117
A.1 Antecedentes . . . 118
A.2 Objetivos . . . 120
A.3 Metodología . . . 121
A.4 Resultados . . . 123
A.5 Conclusiones . . . 125

Figures

3.1 Our method regresses soft-tissue dynamics for parametric avatars. Here we see five different body shapes performing a running motion, each of them enriched with soft-tissue dynamics. We depict the magnitude of the regressed displacements using colormaps (right). 16

3.2 Runtime pipeline of our approach. First, the temporal motion data is encoded in our novel disentangled pose descriptor. Then, the resulting low dimensional vector is concatenated with the skeleton root offsets to form the motion descriptor. This descriptor along with the desired shape parameters are passed through the soft-tissue regressor, which predicts the nonlinear dynamic behaviour of the soft-tissue deformation in a latent space. Finally, the deformation decoder recovers the original full space of deformation offsets for each vertex of the mesh. 17

3.3 Architecture of the multi-modal pose autoencoder. 22

3.4 Result after static pose disentanglement. Our approach effectively removes subject- and shape-dependent features, while retaining the main characteristics of the input pose. 23

3.5 Reconstruction errors of our soft-tissue autoencoder and PCA, for two different body shapes. Notice that our subspace efficiently encodes soft-tissue displacements for parametric shapes, in contrast to previous works [CO18] that required an autoencoder per subject. 26

3.6 Evaluation of generalization to new motions. The sequence `one_leg_jump` was left out at train time, and used only for testing, for subject 50004. We show ground truth meshes and vertex displacements Δ^{GT} (top), and the regressed deformations Δ (bottom). Notice how the magnitude of the regressed displacement closely matches the ground truth. 27

3.7 We quantitatively evaluate the generalization to new shapes of our regressor by looking at the mean vertex speed of the predicted soft-tissue offsets in unposed state in two test sequences. Our model (pink) produces a higher range of dynamics, with large velocities for obese subjects (shape parameter -2.5) and small velocities for thin subjects (shape parameter 0.5). In contrast, previous works (Dyna, in dark blue) produce a much smaller range, resulting in limited generalization capabilities to new subjects. Furthermore, here we also demonstrate that all components of our method contribute to getting the best generalization capabilities. 28

3.8 Sample frames of soft-tissue regression on two test sequences and two test subjects. Colormap depicts the magnitude of the regressed deformation. Notice how our method successfully regresses larger deformations on highly dynamic poses such as in the middle of a jump or when a foot steps on the ground. 29

4.1 Given a garment (left), we learn a deformation model that enables virtual try-on by bodies with different shapes and poses (middle). Our model produces cloth animations with realistic dynamic drape and wrinkles at 250 fps (right). 32

4.2 Overview of our preprocessing and runtime pipelines. As a preprocess, we generate physics-based simulations of multiple animated bodies wearing the same garment. At runtime, our data-driven cloth deformation model works by computing two corrective displacements on the unposed garment: global fit displacements dependent on the body's shape, and dynamic wrinkle displacements dependent on the body's shape and pose. Then, the deformed cloth is skinned on the body to produce the final result. 34

4.3 For tight clothing, data-driven cloth deformations may suffer from apparent collisions with the body (left). We apply a simple postprocessing step to push colliding cloth vertices outside the body (right). 36

4.4 Results of garment fit regression for different bodies. 37

4.5 Results of garment wrinkle regression for different poses. 38

4.6 Quantitative evaluation of generalization to new shapes, comparing our method to retargeting techniques [LCT18; Pon*17]. The top plot shows the error as we increase the body shape to values not used for training, and back, on a static pose (see Figure 4.9). The bottom plot shows the error as we change both the body shape and pose during a test sequence not used for training. 41

4.7 Quantitative evaluation of generalization to new poses, comparing our method to Linear Blend Skinning (LBS) and Linear Regression (LR). 42

4.8 Our nonlinear regression method succeeds to retain the rich and history-dependent wrinkles of the physics-based simulation. Linear regression, on the other hand, suffers blending and smoothing artifacts even on the training sequence shown in the figure. 42

4.9 Our method matches qualitatively the deformations of the ground-truth physics-based simulation when changing the body shape beyond training values. In particular, notice how the T-shirt achieves the same overall drape and mid-scale wrinkles. Retargeting techniques [LCT18; Pon*17], on the other hand, scale the garment, and suffer noticeable artifacts away from the base shape. . 43

4.10 Comparison between DRAPE [Gua*12] (left) and our method (right). DRAPE cannot realistically cope with shape variations, and it is limited to scaling the garment to fit the target shape. In contrast, our method predicts realistically how a garment fits avatars with very diverse body shapes. 44

4.11 Comparison between ClothCap [Pon*17] (left) and our method (right). In ClothCap, the original T-shirt (top-left) is obtained using performance capture, and then scaled to fit a bigger avatar. While the result appears plausible for certain applications, it is not suited for virtual try-on. In contrast, our method produces pose- *and* shape-dependent drape and wrinkles, thus enabling a virtual try-on experience. 44

4.12 Cloth animation produced by our data-driven method on a test sequence. . . . 44

4.13 Comparison between a ground-truth physics-based simulation (top) and our data-driven method (bottom), on a test sequence not used for training (01_01 from [CMU]). Even though our method runs three orders of magnitude faster, it succeeds to predict the overall fit and mid-scale wrinkles of the garment. . 45

5.1 Existing learning-based methods for garment deformations (top) use supervised training schemes that require the expensive computation of large datasets. In contrast, our approach SNUG (bottom) is a learning-based method that enables the self-supervised training of dynamic neural 3D garments, without requiring any ground-truth data. 48

5.2 Overview of our method. First, the recurrent regressor predicts per-vertex offsets as a function of body shape and motion. These offsets are added to the garment template which is then skinned to produce the final result. We train the network by optimizing a set of physical properties of the predicted garments, removing the need for ground-truth data. 50

5.3 The material model used is crucial to obtain realistic garment behaviors. We formulate our losses using the Saint Venant Kirchoff (StVK) model, in contrast to simpler alternatives that lead to less expressive deformations. 54

5.4 Quantitative evaluation of our approach. We evaluate the error in the physics-based terms used in our loss, in the test sequence `01_01` of AMASS [Mah*19]. Sudden motion changes (*e.g.*, jumps) naturally produce peaks in the inertial term, due to drastic changes in the velocity of the garment. Intuitively, cloth dynamics arise when the garment resists those changes induced by the body, therefore lower inertial values indicate that our model learns time-dependent effects better than PBNS [BME21]. 56
5.5 Qualitative comparison with state-of-the-art methods. SNUG generalizes well to unseen body shapes and motions and produces detailed folds and wrinkles. The results of SNUG are on par with the realism of *supervised* methods that require large datasets [SOC19; PLP20] and close to *offline physics-based* simulation [NSO12]. 59
5.6 When trained using same motions and same architecture, direct supervision at the vertex level leads to smoothing artifacts (a). In contrast, our physics-based loss is able to learn more realistic details (b), as shown in this frame from a test sequence. 59
5.7 Qualitative results of our self-supervised method, in validation body shapes and poses unseen during training. SNUG successfully learns highly-realistic garment deformations, including fine wrinkles, as a function of body shape and motion. 60

6.1 Our data-driven method regresses deformed garments via a generative model that is trained to avoid collisions. 62
6.2 Overview of our preprocessing (top) and runtime pipelines (bottom). The decoder network is trained to avoid collisions in a self-supervised fashion, and then employed by the regressor network to reproduce these states at runtime. 64
6.3 Unposing of a T-shirt and a dress in challenging poses: (a) input mesh; (b) unposing with constant weights [PLP20; SOC19], notice the collisions; (c) unposing with variable weights assigned with nearest vertex, it avoids collisions but introduces skinning artifacts and is not temporally stable; (d) unposing with our optimization. 68
6.4 Number of body-garment collisions, evaluated in a test set, during the training of the generative subspace. Our novel self-supervised term, described in Equation 6.16, is key to reduce collisions in unseen sequences. 71
6.5 Fixing collisions as a postprocess can introduce undesired bulges, see chest area in (b). 75
6.6 Generalization to new shapes. Interpolation between two unseen body shapes (left and right) from the AMASS dataset [Mah*19]. Our deshaped canonical space avoids collisions even in shapes far from the training data. 76

6.7 Generalization to new motions. Qualitative comparison with physical simulation [NSO12] (top) in sequence 01_01. Our model (bottom) synthesizes highly realistic dynamics and wrinkles even for challenging unseen motions. 76

7.1 Overview of ULNeF. 83
7.2 Pipeline of our method for mix-and-match virtual try-on. We first preprocess a dataset of garments by simulating each of them in a variety of human shapes. Then, we transform garments into a canonical space, and learn shape-dependent explicit and implicit models. At runtime, we infer explicit and implicit shape-dependent garment deformations, use ULNeF to untangle the implicit representations, and optimize the explicit surfaces to fit into the resulting untangled fields. 86
7.3 Qualitative ablation study of our implicit garment model described in Section 7.3.1. For this particular figure, we use marching cubes to extract the surface. 92
7.4 Given a set of garments (left insets), existing virtual try-on methods [SOC19] infer their fit into a target body shape but produce a heavily entangled results (left). In contrast, ULNeF untangles the garments by directly projecting their neural fields into a collision-free configuration. Since ULNeF allows to specify the desired order, different outfits can be created (center and right). . . 93

8.1 Screenshot of our interactive mix-and-match demo. Despite being limited in scope, this demo entails significant technical challenges that state-of-the-art methods cannot address. The left view represents the results obtained after doing mix-and-match of state-of-the-art data-driven models, which are trained per garment but cannot be mixed together. The right view shows the results obtained with our method for efficient contact resolution, which handles highly challenging cases at interactive frame rates. 96

A.1 Ejemplo de las deformaciones de ropa que podemos generar, en cuestión de milisegundos, gracias a los métodos desarrollados en esta tesis. 117
A.2 Captura de pantalla de nuestra aplicación interactiva. Esta aplicación conlleva importantes retos técnicos que los métodos del estado del arte (izquierda) no son capaces de resolver. Nuestros modelos de ropa y contacto (derecha), son capaces de gestionar casos de alta complejidad en cuestión de milisegundos. 125

Tables

3.1 Reconstruction error of our soft-tissue autoencoder and PCA evaluated in the full test dataset. The autoencoder (AE) performs better than the linear approach (PCA) in all tested subspace sizes. 26

4.1 Per-frame execution times of our method, with and without collision postprocessing. Full physics-based simulation times are also provided for reference. . 45

5.1 To quantitative evaluate our method we compute the physics-based loss terms of our trained model, in unseen sequences, and compare to PBNS. We produce lower errors in all terms, indicating that our approach results in deformations that better match physics-based simulators. 57

5.2 Quantitative ablation study. Each term of our loss contributes to the accuracy of the final result. 57

5.3 Timings, memory requirements, and performance of state-of-the-art methods. Our self-supervised approach avoids the expensive cost of data generation, while also achieving significantly lower training times. 58

6.1 Average number of collisions in 105 test motions from the AMASS dataset [Mah*19]. 74

6.2 Quantitative evaluation of our approach in 5 test sequences and 17 body shapes. 75

6.3 Execution time of each step of our model. 77

6.4 Evaluation time of the networks required to avoid body-garment collisions (*i.e.*, evaluating the diffused body model to project vertices from canonical to pose space) *vs.* the postprocessing time for [PLP20] and [SOC19] using authors' implementation. 77

7.1 Preprocessing time per garment. 88

7.2 Ablation study of the different aspects of our implicit surface model. 91

7.3 Comparison of runtime performance of the main components of ULNeF. We use the authors' implementation to compare the performance of the untangling operator, and an efficient GPU reimplementation to compare the fields. This comparison was conducted in a regular desktop PC equipped with an AMD Ryzen 7 2700 CPU, an Nvidia GTX 1080 Ti GPU, and 32GB of RAM. . . . 92

1 Introduction

Clothing has been an important aspect of human societies throughout history. From a functional perspective, clothing provides a barrier between our skin and the environment that protects us from weather conditions and external hazards. But clothing can be much more than that. The things we wear also serve as a form of self-expression and a way of presenting ourselves to the world. In fact, this desire to have clothes that reflect our individuality is one of the main drivers of the fashion industry, which creates thousands of new garments each season to accommodate a wide range of body shapes and styles.

When buying new clothes, the fit and the style of a garment are the main aspects that influence our decisions. A good fit provides increased levels of comfort and enhances the natural shape of our body, while the style is up to the customer to evaluate depending on their personal taste or the context in which they will wear the garments. Moreover, the way a garment matches other clothes in our wardrobe is also a relevant aspect when choosing what to wear.

Currently, the most reliable way of deciding if a garment suits us is by trying it on our body, which is what we will refer to as *physical* try-on. For many people there is also an undeniable charm in shopping for clothes physically that goes beyond testing the fit and style of the products: it provides an opportunity to socialize with other people, and it allows them to interact with the garments and see how they look in motion. Nevertheless, physical try-on also has several limitations: it requires physical access to garments that may not be available at the moment of visiting the store (or may be available in a limited number of sizes), it is time consuming due to queues and the small number of garments that can be brought to the fitting room, and it does not provide a convenient way of checking how a garment matches other clothes that we may have at home or even clothes from other stores.

In recent years, there has been a growing interest in developing virtual alternatives that combine the reliability of physical try-on with the convenience of online shopping. We will refer to these alternatives as *virtual* try-on. In essence, a virtual try-on system needs to predict how a selection of garments will look on the user without requiring any kind of physical interaction. For such system to be successful, it is imperative that it makes accurate

predictions, provides results with minimal delay, and scales well to thousands or even millions of garments. The goal of this thesis is to develop new methods for virtual try-on that satisfy these requirements. In the following sections, we introduce the steps involved in a virtual try-on application, the associated technical challenges, and the contributions of this thesis toward overcoming them.

1.1 Virtual try-on

In its simplest form, a virtual try-on application involves these steps: first, the user provides body shape information (*e.g.*, images or measurements) and selects a combination of garments to try on, then the application predicts how the garments will fit the user's body and presents the results. The idea of implementing such application has been floating around the retail industry for many years, yet despite significant efforts and numerous prototypes, there are still significant technical challenges that prevent the widespread use of existing implementations. To better understand these challenges let us first define the desirable properties of any virtual try-on system:

- **Accuracy.** To be useful, a virtual try-on system has to provide accurate estimations of the fit of a garment when worn by the user. The system also has to be accurate at conveying the style of the garment and the visual properties of the fabric.

- **Interactivity.** To be enjoyable, a virtual try-on system should provide results with minimal delay and let the user try combinations of garments in an interactive manner. Additional interactivity through animated garments can also greatly enhance the virtual try-on experience.

- **Scalability.** To be cost-effective, the cost of predicting the results and the effort of adding new garments to the system have to be as low as possible. The system also has to scale to a wide range of body shapes and an almost limitless combination of garments.

Currently, no method satisfies all these requirements simultaneously. For example, physics-based approaches [KJM08; Sel*09; NSO12; Cir*14] perform cloth simulations to predict the fit of a garment on a certain body, but the accuracy of these methods comes at the expense of interactivity (the user has to wait for the simulation) and scalability (the simulation requires significant computational resources per user). There is ongoing research to develop simulation methods that satisfy the performance needs of virtual try-on [Tan*18].

Meanwhile, image-based methods [SM06; Zho*12; HSR13; HFE13; Han*18; CML21] formulate virtual try-on as an image synthesis problem, in which the goal is to obtain a new image of the user wearing the selected garments. Working in image space allows these methods to leverage the extensive literature in image synthesis and computer vision, but enforcing physical constraints in the image domain is a challenging problem that greatly hinders the accuracy of the results. Moreover, image-based approaches rely heavily on pictures of professional models wearing the garments, which introduces a bias toward body shapes that are not representative of the full population. As a result, while the synthesized images may convey the style of the outfit, they lack accuracy in the estimation of the fit and struggle with non-average body shapes.

To overcome these issues, this thesis addresses virtual try-on as a 3D problem and builds upon existing 3D human models that capture the diversity of the human body [Lop*15], and physics-based cloth simulation methods that provide accurate fit estimations [WOR11; NSO12].

1.2 Open problems

This section introduces the main open problems for building virtual try-on applications based on 3D avatars and clothing.

Dynamic soft-tissue deformations

Soft-tissue dynamics are fundamental to produce compelling human animations. Most existing methods capable of generating highly dynamic soft-tissue deformations are based on physics-based approaches. However, these methods are challenging to implement due to the inner complexity of the human body, and the expensive simulation process needed to animate the model. Alternatively, data-driven models can potentially learn human soft-tissue deformations as a function of body pose directly from real-world data (*e.g.*, 3D reconstructed sequences). However, in practice, this is a very challenging task due to the highly nonlinear nature of the dynamic deformations, and the scarcity of datasets with sufficient reconstruction fidelity. In this thesis we explore the use of learning-based methods to generate highly expressive soft-tissue dynamics, and address the challenges involved in learning models that generalize well despite the limited training data.

Accurate and fast garment simulation

Cloth simulation is a mature field that is widely used in film productions to create cloth animations that *look* realistic. For virtual try-on though, the appearance of realism is not enough. The simulations need to capture the real behavior of the garments, since failure to do so may result in unhappy customers and returned orders. Some works in cloth simulation [WOR11; Mig*12] address this challenge by extracting measurements from real pieces of fabrics and tuning simulation parameters accordingly. Other methods go as far as to simulate cloth at the yarn level [KJM10; Cir*14; CLO15] in order to capture mechanical behaviors that cannot be replicated with thin-shell models. Unfortunately, there is a significant tradeoff between the realism of a simulation and the computational cost of running it, and current solutions do not meet the demanding requirements of virtual try-on.

Narrowing this tradeoff between accuracy and performance is one of the goals of this thesis. Our main insight is that virtual try-on is a highly constrained subproblem of cloth simulation in which the garment deformation can be modeled directly as a function of body parameters (*e.g.*, shape, pose). Since learning-based models are capable of approximating complex functions when there is a strong correlation between inputs and outputs, we propose using machine learning techniques to perform accurate fit predictions at a fraction of the cost of traditional cloth simulators.

Mix and match virtual try-on

In addition to estimating the fit of a garment, a virtual try-on system should also let the user mix and match different garments to create new outfits. Mix-and-match virtual try-on requires finding a collision-free configuration of the garments chosen by the user but, unfortunately, there are no automatic and robust tools to address this task. Traditional cloth simulators rely on a collision-free initial configuration and use continuous collision detection to prevent garments from reaching a configuration with collisions, but the initial collision-free state is usually obtained manually using 3D editing tools.

Another challenge in mix and match virtual try-on is the inherent ambiguity in solving garment collisions, since there is not a unique solution for how the garments should be separated. For example, a shirt may be tucked inside the pants or it may be hanging out, so automatic solutions to this problem also need to account for the user's intent.

Recently, Buffet *et al.* [Buf*19] have proposed a novel method to untangle layered garments that relies on implicit surface representations. The goal of the method is to obtain a collision-free configuration that can be fed to a cloth simulator, but the process can take several minutes to compute. We seek to improve this method by bringing its computational cost closer to the requirements of virtual try-on. To do so, we drive our attention to neural fields [Xie*22], which in the last few years have arisen as a powerful tool to efficiently model implicit surfaces.

1.3 Contributions and publications

These are the contributions of this thesis toward solving the open problems presented in the previous section:

- A learning-based method to model realistic soft-tissue dynamics as a function of body shape and motion. At the core of our method there are three key contributions that enable us to model highly realistic dynamics and achieve better generalization capabilities than state-of-the-art methods, while training on the same data. First, a novel motion descriptor that disentangles the standard pose representation by removing subject-specific features; second, a recurrent neural network that generalizes to unseen body shapes and motions; and third, a highly efficient nonlinear deformation subspace capable of representing soft-tissue deformations of arbitrary bodies. (Chapter 3)

- A learning-based method to produce detailed clothing deformations at interactive frame rates. Our method is built upon standard skinning techniques, which we use to obtain an approximate model of the garment's motion. We then enhance this model by introducing a vector of corrective offsets that are computed by a recurrent neural network. In order to obtain realistic animations, the network learns these offsets from physically simulated sequences. (Chapter 4)

- A self-supervised method to learn clothing animations without requiring ground-truth simulations. Our key contribution is to realize that physics-based deformation models, traditionally solved on a frame-by-frame basis by implicit integrators, can be recast as an optimization problem. We leverage such optimization-based scheme to formulate a set of physics-based loss terms that can be used to train neural networks without precomputing ground-truth data. This allows us to learn models for interactive garments, including dynamic deformations and fine wrinkles, with a speed-up of two orders of magnitude in training time compared to supervised approaches. (Chapter 5)

- A generative model for 3D garment deformations that enables us to learn, for the first time, a data-driven method for virtual try-on that effectively addresses garment-body collisions. In contrast to existing methods that require an undesirable postprocessing step to fix garment-body interpenetrations at test time, our approach directly outputs 3D garment configurations that do not collide with the underlying body. Key to our success is a new canonical space for garments that removes pose-and-shape deformations already captured by a new diffused human body model, which extrapolates body surface properties such as skinning weights and blendshapes to any 3D point. We leverage this representation to train a generative model with a novel self-supervised collision term that learns to reliably solve garment-body interpenetrations. (Chapter 6)

- A novel method to untangle layered garments that enables mix-and-match virtual try-on at interactive framerates. To this end, we propose a neural model that untangles layered neural fields to represent collision-free garment surfaces. The key ingredient is a neural untangling projection operator that works directly on the layered neural fields, not on explicit surface representations. (Chapter 7)

These contributions have led to the following publications:

- Igor Santesteban, Miguel A. Otaduy, and Dan Casas. "Learning-Based Animation of Clothing for Virtual Try-On". *Computer Graphics Forum (Proc. Eurographics)* (2019)

- Igor Santesteban, Elena Garces, Miguel A. Otaduy, and Dan Casas. "SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Humans". *Computer Graphics Forum (Proc. Eurographics)* (2020)

- Igor Santesteban, Nils Thuerey, Miguel A. Otaduy, and Dan Casas. "Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On". *Proc. of Computer Vision and Pattern Recognition (CVPR)* (2021)

- Igor Santesteban, Miguel A. Otaduy, and Dan Casas. "SNUG: Self-Supervised Neural Dynamic Garments". *Proc. of Computer Vision and Pattern Recognition (CVPR)* (2022)

- Igor Santesteban, Nils Thuerey, Miguel A. Otaduy, and Dan Casas. "ULNeF: Untangled Layered Neural Fields for Mix-and-Match Virtual Try-On". *Under review*

2 Background

Virtual avatars and virtual garments are the main pillars of a virtual try-on application. Over the years, both topics have attracted great interest from the computer graphics and computer vision communities, and this interest has led to a wide range of excellent publications that serve as the foundation for this thesis. This chapter presents an overview of these works organized as follows:

- Section 2.1 reviews the literature on human body modeling and estimation of accurate avatars of real people. We also discuss the different approaches to model soft-tissue deformations and their importance in creating realistic human animations.
- Section 2.2 reviews the literature on garment modeling and design. We also discuss the existing methods to predict cloth deformations as well as their advantages and limitations.

2.1 Virtual avatars

While existing technologies are capable of producing accurate digitizations of real people, achieving high levels of accuracy often requires the use of expensive multi-camera setups or markers [SH07; Vla*08; Vla*09; NH14]. Ongoing research aims to make this process more accessible by removing the need for markers and inferring 3D surfaces directly from single-view RGB images [Sai*19; Sai*20; Zha*21b]. While these methods can be used to create subject-specific 3D avatars, in this thesis we drive our focus towards parametric human models, which are capable of representing a wide range of body shapes and poses in a low-dimensional parameter space.

2.1.1 Parametric human models

Pioneering data-driven human models interpolate manually sculpted static 3D meshes to generate new samples [SRC01]. With the development of laser scanning technologies,

capable of reconstructing 3D static bodies with great level of detail, the data-driven field became popular. Hilton *et al.* [HSC02] automatically fit an skeleton to a static scan to generate animated characters. Allen *et al.* proposed one of the first methods to model upper body [ACP02] and full body [All*03] deformations using a shape space learned from static scans and an articulated template. Anguelov *et al.* [Ang*05] went one step further and modeled both shape *and* pose dependent deformations directly from data. Many follow-up data-driven methods have appeared [Has*09; Jai*10; Hir*12; CLZ13; Yan*14; FCS15; ZB15; Lop*15; Pis*17].

In this thesis we use the popular SMPL human model [Lop*15]. SMPL encodes bodies by deforming a rigged human template according to shape and pose-dependent deformations that are learned from data. Subsequent works use a similar approach to model hands [RTB17], faces [Li*17b], and bodies with expressive hands and faces [Pav*19]. The parameter space of SMPL provides a compact representation for body shapes and poses, and is compatible with large datasets of motion capture data [Mah*19]. Moreover, there is also a significant body of work on fitting model parameters to pictures of real people [Bog*16; Kan*18; Omr*18; Pav*19; Fen*21], a highly relevant problem in the context of virtual try-on. Using a parametric model greatly reduces the complexity of the solution space, which is key for estimating avatars from ambiguous inputs such as monocular images.

2.1.2 Soft-tissue deformation

Soft-tissue dynamics are a key ingredient of realistic human animations and existing works for modeling soft-tissue deformations can be categorized into two main trends: data-driven models, which learn deformations directly from data; and physically-based models, which compute body deformations by solving a simulation problem, usually consisting of a kinematic model coupled with a deformable layer.

Data-driven models

Initial works in data-driven soft-tissue deformation used *sparse* marker-based systems to acquire data. The pioneering work of Park and Hodgins [PH06] reconstructs soft-tissue motion of an actor by fitting a 3D mesh to 350 tracked points. In subsequent work [PH08], they proposed a second-order dynamics model to synthesize skin deformation as a function of body motion. Similar to the method presented in Chapter 3, they represent both body pose and dynamic displacements in a low-dimensional space. However, their method does

not generalize to different body shapes. Neumann *et al.* [Neu*13a] also used sparse markers to capture shoulder and arm deformations of multiple subjects in a multi-camera studio. They were able to model muscle deformations as a function of shape, pose, and external forces, but their method is limited to the shoulder-arm area, and cannot learn temporal dynamics. Similarly, Loper *et al.* [LMB14] did not learn dynamics either, but they were able to estimate full body pose *and* shape from a small set of motion capture markers. Remarkably, despite their lack of explicit dynamics, their model can reproduce soft-tissue motions by allowing body shape parameters to change over time.

More recently, 3D/4D scanning technologies and mesh registration methods [Bra*08; CBI10; Dou*15; Bog*17; Rob*17; Pon*17] allow the reconstruction of high-quality dynamic sequences of human performances. These techniques have paved the way for data-driven methods that leverage *dense* 3D data, usually in the form of temporally coherent 3D mesh sequences, to extract deformation models of 3D humans. Neumann *et al.* [Neu*13b] used 3D mesh sequences to learn sparse localized deformation modes, but did not model temporal dynamics. Tsoli *et al.* [TMB14] reconstructed 3D meshes of people breathing in different modes, and built a statistical model of body surface deformations as a function of lung volume. Casas and Otaduy [CO18] modeled full-body soft-tissue deformations as a function of body motion using a neural-network-based nonlinear regressor. Their model computes per-vertex 3D offsets encoded in an efficient subspace, however, it is subject-specific and does not generalize to different body shapes. Closest to our work is Dyna [Pon*15], a state-of-the-art method that relates soft-tissue deformations to motion and body shape from 4D scans. Dyna uses a second-order auto-regressive model to output mesh deformations encoded in a subspace. Despite its success in modeling surface dynamics, we found that its generalization capabilities to unseen shapes and poses are limited due to the inability to effectively disentangle pose from shape and subject style. Furthermore, Dyna relies on a linear PCA subspace to represent soft-tissue deformations, which struggles to reproduce highly non-linear deformations.

DMPL [Lop*15] proposes a soft-tissue deformation model heavily inspired in Dyna, with the main difference that it uses a vertex-based representation instead of triangle-based. However, DMPL suffers from the same limitations as Dyna mentioned above. In Chapter 3 we also propose a vertex-based representation, which eases the implementation in standard character rigging pipelines, while achieving superior generalization capabilities and more realistic dynamics.

Physically-based models

The inherent limitation of data-driven models is their struggle to generate deformations far from the training examples. Physically-based models overcome this limitation by formulating the deformation process within a simulation framework. However, these approaches are not free of difficulties: defining an accurate and efficient mechanical model to represent human motions, and solving the associated simulations is hard.

Initial works used layered representations consisting of a deformable volume for the tissue layer, rigidly attached to a kinematic skeleton [Cap*02; LCA05]. Liu *et al.* [Liu*13] coupled rigid skeletons for motion control with a pose-based plasticity model to enable two-way interaction between skeleton, skin, and environment. McAdams *et al.* [McA*11] showed skin deformations with a discretization of corotational elasticity on a hexahedral lattice around the surface mesh, but did not run at real-time rates. Xu and Barbič [XB16] used secondary Finite Element Method (FEM) dynamics and model reduction techniques to efficiently enrich the deformation of a rigged character. To speed up simulations, Position-Based Dynamics (PBD) [BMM17] solvers have been widely used for different physics systems, also for human soft tissue [DB13; KB18] and muscle deformation [RMS19]. Projective Dynamics, another common approach to accelerate simulations, has also been used for simulating deformable characters [LLK19]. Meanwhile, Pai *et al.* [Pai*18] presented a novel hand-held device to estimate the mechanical properties of real human soft-tissue. More recently, Romero *et al.* [Rom*20] proposed a hybrid method that models soft-tissue deformations as a combination of a data-driven statistical model and an FEM simulation.

2.2 Virtual garments

2.2.1 Design and modeling

Designing a garment is a time-consuming process that usually starts with a sketch of the desired outcome, and is followed by the creation of 2D patterns. A pattern is a set of flat panels (*i.e.*, patches of fabric) that are sewn together to create a garment. The size and shape of the panels are key to provide a good fit to the wearer but, due to the wide range of body shapes, a single pattern cannot fit all customers. To overcome this problem, most retail stores use a sizing system to adapt the patterns to a small but diverse subset of bodies, a process that is done manually by a garment designer through trial and error. The customers can then take measurements of their own bodies and refer to a sizing chart to see which size

is adequate for them, but in practice, this approach is not entirely reliable. This is because a discrete set of 5-6 sizes is not enough to provide a good fit to all potential customers, and each brand uses its own sizing system (*i.e.*, an M sized shirt from one brand may fit perfectly while the same size from another brand may be ill-fitting). As a result, most customers rely on physical try-on to assess if a garment is indeed suitable for them.

Currently, most of the steps from the conception of a garment to its fabrication and distribution involve manual labor, but there are ongoing efforts toward the digitalization of the fashion industry. While virtual try-on is one example of such efforts, there has also been a surge of digital tools for garment design (*e.g.*, Optitex, Marvelous Designer), garment capture [Sch*05; WCF07; Bra*08; Pon*17], automatic garment adjustment [Bar*16; Wan18; Wol*21], and even methods to create garments directly from sketches [Li*17a; Wan*18]. Despite addressing different problems, all these methods require estimating cloth deformations in one way or another. The following section provides an overview of existing approaches to address this task.

2.2.2 Cloth deformation

Existing methods to model how cloth and garments deform can be categorized into two groups: physics-based models and data-driven models.

Physically-based models

Physics-based simulation methods use discretizations of classical mechanics to model how cloth deforms, and typically comprise three steps: computation of internal forces, collision detection, and collision response [Nea*06]. These methods produce highly-realistic simulations, generalize to different garments, and can handle body-garment collisions, however, they fail to meet the combined robustness and performance needed for real-time applications such as virtual try-on.

A wide range of strategies have been proposed to address the computational bottleneck in physics-based methods. Recent attempts include approximations of the dynamics to trade physical accuracy for speed [Ben*14; Bou*14; Ly*20], adaptive remeshing to refine surface discretization [Lee*10; NSO12], upsampling details to enrich coarse simulations [Kav*11; ZBO13], and GPU-based solvers [Tan*16; FTP16; Tan*18]. Moreover, while the majority of the cloth simulation models represent the fabric as a continuum, some works use

yarn-level representations for high-resolution detail [KJM08; Cir*14] and propose efficient representations to handle contact between yarns [CLO15].

Another challenge in physics-based simulation is the estimation of the model parameters. To this end, some works measure the deformation of small pieces of fabrics under controlled setups, and tune simulation parameters to match the real samples [WOR11; Mig*12]. Alternative methods attempt to recover material parameters directly from videos by a model fitting process [Bha*03; Sto*10; Mon*12] or learn this task directly from data [Bou*13; Wu*16; YLL17; Ras*20; Run*20]. Despite the impressive progress towards addressing the critical points in physics-based models, virtual try-on applications require faster and easier to set up methods.

Data-driven models

In contrast to physics-based models, which typically require solving large systems of nonlinear equations at each time step, learning-based methods aim at estimating a single function that directly outputs the desired deformation for any input. Inspired by early works on Pose Space Deformation [LCF00], a common strategy is to learn parametric garment deformations, which are added to a mesh template, as a function of pose [Gua*12; Wan*19], shape [Vid*20], pose-and-shape [SOC19; BME20], design [PLP20; Wan*18; Ma*20], or garment size [Tiw*20].

To this end, state-of-the-art methods for garments use *supervised* strategies that require large datasets of ground-truth data of the specific task to be learned. This methodology has been recently explored for many use cases, including 3D reconstruction [All*19; All*18; Sai*19; Zhu*20], garment design [SLL20; Vid*20; Wan*18], animation [Ber*21; Hua*20; Wan*19; PLP20; Gun*19; Ma*20], and virtual try-on [Zha*21b; Bha*19; SOC19; Gua*12]. To efficiently tackle the learning task, and depending on the goal of each method, different supervision terms and domains have been used. Most methods use direct 3D supervision at the vertex level [SOC19; PLP20; Vid*20; Gun*19], but image-based 2D supervision in form of UV maps [LCT18; SLL20; Jin*20], point clouds [Sai*21; Ma*21], or sketches [Wan*18] also exist. Very recently, implicit representations have shown impressive results on learning to deform humans [Den*20; Mih*21; AXS21] and dress avatars [Sai*21; Tiw*21; Cor*21; Wan*21].

Datasets are a fundamental piece to enable supervision, and most methods [SOC19; PLP20; Wan*18; BME20] opt for synthetic data generated with physics-based simulators such as ARCSim [NSO12] or Argus [Li*18]. The methods presented in Chapters 4 and 6 belong

to this category. Alternatively, other methods [LCT18; Tiw*20; Ma*20; Sai*21] use high-quality 3D scans obtained in expensive multi-camera setups [Zha*17; Pon*17]. Despite the success of all these supervised methods for learning-based garments, relying on ground-truth data to train the models is a major limitation due to the associated costs and challenges in the data acquisition process.

Self-supervised strategies are the ideal alternative to circumvent the need for ground-truth data in learning-based methods [SE17]. Instead of relying on losses that evaluate prediction error based on the difference with respect to ground-truth samples, *self-supervised* methods use implicit properties of the training data (or domain) as a supervision signal [Zhu*19]. This strategy is nowadays very popular in data-driven methods for image-based problems [Zhu*17; Li*20a; Raj*18], however, almost all state-of-the-art approaches for learning 3D garment deformations rely on ground-truth data [PLP20; SOC19; Gun*19]. For 3D deformations tasks not related to garments, many works use physics laws or constraints as a supervision signal [Zhu*19; Tom*17; Xie*18]. For example, Tompson *et al.* [Tom*17] enforce incompressibility constraints to learn to solve the system of equations required in physics-based fluid simulation, Xie *et al.* [Xie*18] enforce temporal coherence of consecutive frames in fluid simulations to enhance detail, and Zhu *et al.* [Zhu*19] incorporate the governing equations of the physical model (*i.e.*, Partial Differential Equations, PDEs) in the loss to learn image-based flow simulations.

Despite the significant progress in self-supervised learning, no previous work addresses the learning of 3D garments in a self-supervised manner, with just the notable and very recent exception of PBNS [BME21]. PBNS proposes to learn pose space deformations for garments by enforcing *static* physical consistency during the training of the model. In Chapter 5 we follow a similar underlying idea, but propose to use a full physics-based deformation scheme recast as an optimization problem to learn, for the first time, a model for *dynamic* garment deformations with self-supervision only. Additionally, our approach learns shape-dependent effects and is able to cope with a material model that produces more realistic and finer wrinkles.

Image-based models

Virtual try-on has also been approached from an image-based point of view. Image-based methods aim to generate compelling 2D images of dressed people, without dealing with any 3D model or simulation of any form. Hilsmann *et al.* [HFE13] proposed a pose-dependent image-based method that interpolates between images of clothes. More recently, Han *et al.* [Han*18] presented a learning-based method that achieves photorealistic results

using convolutional neural networks. Subsequent works further improve the quality of the synthesized images [Lee*19; YWX19; Han*19; Yan*20; Ge*21a], solve artifacts by reducing the reliance on 2D segmentation [IMC20; Ge*21b], support mix and match virtual try-on [Neu*20; Li*21; CML21], and synthesize images for arbitrary poses [Don*19; Wan*20].

Despite the outstanding progress and the success in generating good-looking images, these methods do not provide accurate information in terms of how a garment fits the user, since they do not account for the size of the garments. Moreover, image-based virtual try-on methods are usually trained and validated in images of professional models under good lighting conditions and white backgrounds, and generalizing to in-the-wild images as well as diverse body shapes is still an unsolved problem.

Although 2D and 3D-based methods have evolved independently from each other, in the last year there has been remarkable progress towards bringing these two lines of research together. The work of Habermann *et al.* [Hab*21] generates realistic 3D avatars with motion-dependent geometry and motion- and view-dependent textures. The method does not require ground-truth 3D garment deformations (*e.g.*, cloth simulations or scans), instead, it learns directly from images obtained in a multi-camera studio. Similarly, Burov *et al.* [BNT21] learn clothed human models from monocular RGB-D sequences that can be used to produce new animations with pose-dependent wrinkles. Meanwhile, Zhao *et al.* [Zha*21b] propose a method that, given an image of the user and another image of a garment, creates a static 3D avatar of the user wearing the garment. The resulting 3D avatar can be rendered from arbitrary points of view, but the accuracy of the fit is still limited by the image-based representation of the garment.

This thesis addresses virtual try-on as a 3D problem, but we hope that this trend of mixing 2D and 3D representations will converge towards hybrid methods that combine the accuracy and interactivity of 3D models with the photorealism and versatility of 2D approaches.

3 Supervised learning of soft-tissue deformations

This chapter presents SoftSMPL, a learning-based method to model realistic soft-tissue dynamics as a function of body shape and motion. Datasets to learn such task are scarce and expensive to generate, which makes training models prone to overfitting. At the core of our method there are three key contributions that enable us to model highly realistic dynamics and achieve better generalization capabilities than state-of-the-art methods, while training on the same data. First, a novel motion descriptor that disentangles the standard pose representation by removing subject-specific features; second, a recurrent neural network that generalizes to unseen shapes and motions; and third, a highly efficient nonlinear deformation subspace capable of representing soft-tissue deformations of arbitrary body shapes. We demonstrate qualitative and quantitative improvements over existing methods and, additionally, we show the robustness of our method on a variety of motion capture databases. The contributions presented in this chapter have led to the following publication:

Igor Santesteban, Elena Garces, Miguel A. Otaduy, and Dan Casas. "SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Humans". *Computer Graphics Forum (Proc. Eurographics)* (2020)

3.1 Introduction

Soft-tissue dynamics are fundamental to produce compelling human animations. Most existing methods capable of generating highly dynamic soft-tissue deformations are based on physics-driven approaches. However, these methods are challenging to implement due to the inner complexity of the human body, and the expensive simulation process needed to animate the model. Alternatively, data-driven models can potentially learn human soft-tissue deformations as a function of body pose directly from real-world data (*e.g.*, 3D reconstructed sequences). However, in practice, this is a very challenging task due to the

highly nonlinear nature of the dynamic deformations, and the scarcity of datasets with sufficient reconstruction fidelity.

In this work, we propose a novel learning-based method to animate parametric human models with highly expressive soft-tissue dynamics. SoftSMPL takes as input the shape descriptor of a body and a motion descriptor, and produces dynamic soft-tissue deformations that generalize to unseen body shapes and motions. Key to our method is to realize that humans move in a highly personalized manner, *i.e.*, motions are shape and subject dependent, and these subject-dependant features are usually entangled in the pose representation.

Previous methods fail to disentangle body pose from shape- and subject-specific features; therefore, they overfit the relationship between tissue deformation and pose, and generalize poorly to unseen body shapes and motions. Our method overcomes this limitation by proposing a new representation to disentangle the traditional pose space in two steps. First, we propose a solution to encode a compact and *deshaped* representation of body pose which eliminates the correlation between individual *static* poses and subject. Second, we propose a motion transfer approach, which uses person-specific models to synthesize animations for pose (and style) sequences of other persons. As a result, our model is trained with data where pose and subject-specific *dynamic* features are no longer entangled. We complement this contribution with a highly efficient nonlinear subspace to encode tissue deformations of arbitrary bodies, and a recurrent neural network as our learning-based animation model. We demonstrate qualitative and quantitative improvements over previous methods, as well as robust performance on a variety of motion capture databases.

Figure 3.1: Our method regresses soft-tissue dynamics for parametric avatars. Here we see five different body shapes performing a running motion, each of them enriched with soft-tissue dynamics. We depict the magnitude of the regressed displacements using colormaps (right).

3.2 Method

Our animation method for soft-tissue dynamics takes as input descriptors of body shape and motion, and outputs surface deformations. These deformations are represented as per-vertex 3D displacements of a human body model, described in Section 3.2.1, and encoded in an efficient nonlinear subspace, described in Section 3.2.2. At runtime, given body and motion descriptors, we predict the soft-tissue deformations using a novel recurrent regressor proposed in Section 3.2.3. Figure 3.2 depicts the architecture of our runtime pipeline, including the motion descriptor, the regressor, and a soft-tissue decoder to generate the predicted deformations.

In addition to our novel subspace and regressor, our key observation to achieve highly expressive dynamics with unprecedented generalization capabilities is an effective disentanglement of the pose space. In Section 3.3, we argue and demonstrate that the standard pose space (*i.e.*, vector of joint angles θ) used in previous methods is entangled with subject-specific features. This causes learning-based methods to overfit the relationship between tissue deformation and pose. In Section 3.3.1 we identify *static* features, mostly due to the particular anatomy of each person, that are entangled in the pose space, and propose a *deshaped* representation to effectively disentangle them. Furthermore, in Section 3.3.2 we identify *dynamic* features that manifest across a sequence of poses (also known as *style*), and propose a strategy to deal with them.

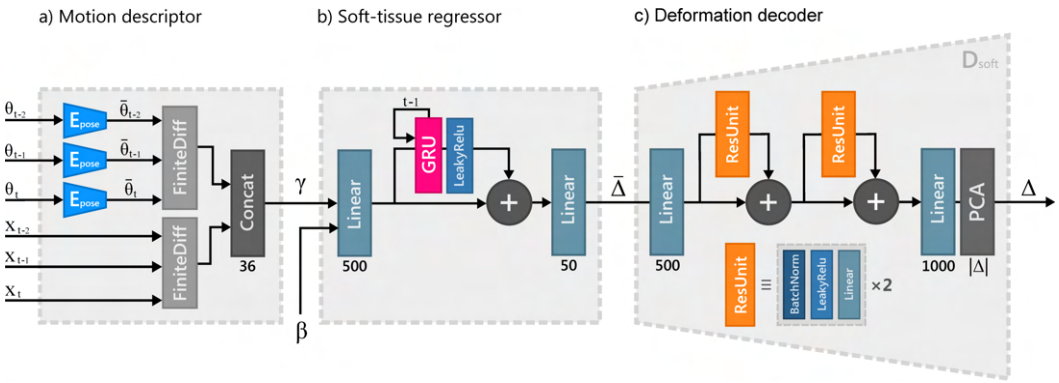


Figure 3.2: Runtime pipeline of our approach. First, the temporal motion data is encoded in our novel disentangled pose descriptor. Then, the resulting low dimensional vector is concatenated with the skeleton root offsets to form the motion descriptor. This descriptor along with the desired shape parameters are passed through the soft-tissue regressor, which predicts the nonlinear dynamic behaviour of the soft-tissue deformation in a latent space. Finally, the deformation decoder recovers the original full space of deformation offsets for each vertex of the mesh.

3.2.1 Human model

We build our soft-tissue model on top of standard human body models (*e.g.*, [FCS15; Lop*15]) controlled by shape parameters $\beta \in \mathbb{R}^{|\beta|}$ (*e.g.*, principal components of a collection of body scans in rest pose) and pose parameters $\theta \in \mathbb{R}^{|\theta|}$(*e.g.*, joint angles). These works assume that a deformed body mesh $M(\beta, \theta) \in \mathbb{R}^{3\times V}$, where V is the number of vertices, is obtained by

$$M(\beta, \theta) = W(T(\beta, \theta), \beta, \theta, \mathcal{W}) \tag{3.1}$$

where $W(\cdot)$ is a skinning function (*e.g.*, linear blend skinning, dual quaternion, etc.) with skinning weights $\mathcal{W}$ that deforms an unposed body mesh $T(\beta, \theta) \in \mathbb{R}^{3\times V}$.

Inspired by Loper *et al.* [Lop*15], who obtain the unposed mesh $T(\beta, \theta)$ by deforming a body mesh template $\mathbf{T} \in \mathbb{R}^{3\times V}$ to incorporate changes in shape $B_{\mathrm{s}}(\beta)$ and pose corrective displacements $B_{\mathrm{p}}(\theta)$, we propose to further deform the body mesh template to incorporate soft-tissue dynamics. More specifically, we define our unposed body mesh as

$$T(\beta, \theta, \gamma) = \mathbf{T} + B_{\mathrm{s}}(\beta) + B_{\mathrm{p}}(\theta) + B_{\mathrm{d}}(\gamma, \beta), \tag{3.2}$$

where $B_{\mathrm{d}}(\gamma, \beta) = \Delta \in \mathbb{R}^{3\times V}$ is a soft-tissue regressor that outputs per-vertex displacements required to reproduce skin dynamics given a shape parameter β and a motion descriptor γ. Notice that, in contrast to previous model-based works that also predict soft-tissue displacements [Pon*15; Lop*15; CO18], our key observation is that such regressing task cannot be formulated directly as function of pose θ (and shape β), because subject-specific information is entangled in that pose space. See Section 3.3 for a detailed description of our motion descriptor γ and full details on our novel pose disentanglement method.

3.2.2 Soft-tissue deformations subspace

We represent soft-tissue deformations Δ as per-vertex 3D offsets of a body mesh $\mathbf{T}$ in an unposed state. This representation allows to isolate the soft-tissue deformation component from other deformations, such as pose or shape.

Given the data-driven nature of our approach, in order to train our model it is crucial that we define a strategy to extract ground truth deformations $\Delta^{\text{GT}} \in \mathbb{R}^{3\times V}$ from real world data. To this end, in a similar spirit to [Pon*15; Lop*15; Pon*17], given a dataset $\mathcal{S} = \{\mathbf{S}_t\}_{t=0}^{T-1}$ of 4D scans with temporally consistent topology, we extract the soft-tissue component of each mesh $\mathbf{S} \in \mathbb{R}^{3\times V}$ as

$$\Delta^{\text{GT}} = W^{-1}(\mathbf{S}, \theta, \mathcal{W}) - \mathbf{T} - B_{\text{P}}(\theta) - B_{\text{S}}(\beta), \tag{3.3}$$

where $W^{-1}(\cdot)$ is the inverse of the skinning function, $B_{\text{P}}(\theta)$ a corrective pose blendshape, and $B_{\text{S}}(\beta)$ a shape deformation blendshape (see [Lop*15] for details on how the latter two are computed). Solving Equation 3.3 requires estimating the pose θ and shape β parameters for each mesh $\mathbf{S}$, which is a priori unknown (*i.e.*, the dataset $\mathcal{S}$ contains only 3D meshes, no shape or pose parameters). Similar to [Pon*15], we solve the optimization problem:

$$\underset{\theta,\beta}{\operatorname{argmin}} ||\mathbf{S} - M(\theta, \beta)||_2 \tag{3.4}$$

to estimate the shape β and pose θ parameters of each scan $\mathbf{S}$ in the dataset $\mathcal{S}$.

Despite the highly-convenient representation of encoding soft-tissue deformations as per-vertex 3D offsets $\Delta \in \mathbb{R}^{3\times V}$, this results in a too high-dimensional space for an efficient learning-based framework. Previous works [Lop*15; Pon*15] use linear dimensionality reduction techniques (*e.g.*, Principal Component Analysis) to find a subspace capable of reproducing the deformations without significant loss of detail. However, soft-tissue deformations are highly nonlinear, hindering the reconstructing capabilities of linear methods. We mitigate this by proposing a novel autoencoder to find an efficient nonlinear subspace to encode soft-tissue deformations of parametric humans.

Following the standard autoencoder pipeline, we define the reconstructed (*i.e.*, encoded-decoded) soft-tissue deformation as

$$\Delta_{\text{rec}} = D_{\text{soft}}(E_{\text{soft}}(\Delta)), \tag{3.5}$$

where $\bar{\Delta} = E_{\text{soft}}(\Delta)$ and $D_{\text{soft}}(\bar{\Delta})$ are encoder and decoder networks, respectively, and $\bar{\Delta} \in \mathbb{R}^{|\bar{\Delta}|}$ soft-tissue displacements projected into the latent space. We train our deformation

autoencoder by using a loss function $\mathcal{L}_{\text{rec}}$ that minimizes both surface and normal errors between input and output displacements as follows

$$\mathcal{L}_{\text{surf}} = \|\Delta - \Delta_{\text{rec}}\|_2 \tag{3.6}$$

$$\mathcal{L}_{\text{norm}} = \frac{1}{F}\sum_{f=1}^{F} \|1 - N_f(\Delta) \cdot N_f(\Delta_{\text{rec}})\|_1 \tag{3.7}$$

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{surf}} + \lambda_{\text{norm}}\mathcal{L}_{\text{norm}} \tag{3.8}$$

where F is the number of faces of the mesh template, $N_f(\Delta)$ the normal of the f^{th} face, and λ_{norm} is set to 1000. Notice that, during training, we use ground truth displacements Δ^{GT} from a variety of characters which enables us to find a subspace that generalizes well to encode soft-tissue displacements of *any* human shape. This is in contrast to previous works [CO18] that need to train shape-specific autoencoders.

We implement the encoder E_{soft} and decoder D_{soft} using a fully-connected neural network architecture composed of several residual units [He*16b]. Inspired by the work of Fulton *et al.* [Ful*19], we initialize the first and last layers of the autoencoder with weights computed using PCA, which eases the training of the network. In Figure 3.2 (right) we depict the decoder D_{soft}. The encoder E_{soft} uses an analogous architecture.

3.2.3 Soft-tissue deformation regressor

In this section we describe the main component of our runtime pipeline: the soft-tissue regressor R, illustrated in Figure 3.2 (center). Assuming a motion descriptor γ (which we discuss in detail in Section 3.3.1) and a shape descriptor β, our regressor outputs the predicted soft tissue displacements $\bar{\Delta}$. These encoded displacements are subsequently fed into the decoder D_{soft} to generate the final per-vertex 3D displacements

$$\Delta = D_{\text{soft}}(R(\gamma, \beta)). \tag{3.9}$$

To learn the naturally nonlinear dynamic behavior of soft-tissue deformations, we implement the regressor R using a recurrent architecture based on Gated Recurrent Units (GRU) [Cho*14]. Recurrent architectures learn which information of previous frames is relevant and which not, resulting in a good approximation of the temporal dynamics. This is in contrast to modeling temporal dependencies by explicitly adding the output of one step as the input of the next step, which is prone to instabilities specially in nonlinear models. Furthermore, our regressor also uses a residual shortcut connection to skip the GRU layer

altogether, which improves the flow of information [He*16a]. We initialize the state of the GRU to zero at the beginning of each sequence.

We train the regressor R by minimizing a loss $\mathcal{L}_{\text{reg}}$, which enforces predicted vertex positions, velocities, and accelerations to match the latent space deformations $\bar{\Delta}$,

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{vel}} + \mathcal{L}_{\text{acc}} \tag{3.10}$$

3.3 Disentangled motion descriptor

To efficiently train the soft-tissue regressor $R(\gamma, \beta)$, described earlier in Section 3.2.3, we require a pose-disentangled and discriminative motion descriptor γ. To this end, in this section we propose a novel motion descriptor. It encompasses the velocity and acceleration of the body root in world space X, a novel pose descriptor $\bar{\theta}$, and the velocity and acceleration of this novel pose descriptor, as follows:

$$\gamma = \{\bar{\theta}, \frac{\mathrm{d}\bar{\theta}}{\mathrm{d}t}, \frac{\mathrm{d}^2\bar{\theta}}{\mathrm{d}t^2}, \frac{\mathrm{d}X}{\mathrm{d}t}, \frac{\mathrm{d}^2X}{\mathrm{d}t^2}\}. \tag{3.11}$$

In the rest of this section we discuss the limitation of the pose descriptors used in state-of-the-art human models, and introduce a new disentangled space $\bar{\theta}$ to remove *static* subject-specific features (Section 3.3.1). Moreover, we also propose a strategy to remove *dynamic* subject-specific features (Section 3.3.2) from sequences of poses.

3.3.1 Static pose disentanglement

The regressor R proposed in Section 3.2.3 relates body motion and body shape to soft-tissue deformations. To represent body motion, a standard parameterization used across many human models [FCS15; Ang*05; LMB14; Lop*15] is the joint angles of the kinematic skeleton, θ. However, our key observation is that this pose representation is entangled with shape- and subject-specific information that hinders the learning of a pose-dependent regressor. Additionally, Hahn *et al.* [Hah*14] also found that using joint angles to represent pose leads to a high-dimensional space with redundancies, which makes the learning task harder and prone to overfitting. We hypothesize that existing data-driven parametric human models are less sensitive to this entanglement and overparameterization because they learn simpler deformations with much more data. In contrast, we model soft-tissue with a limited

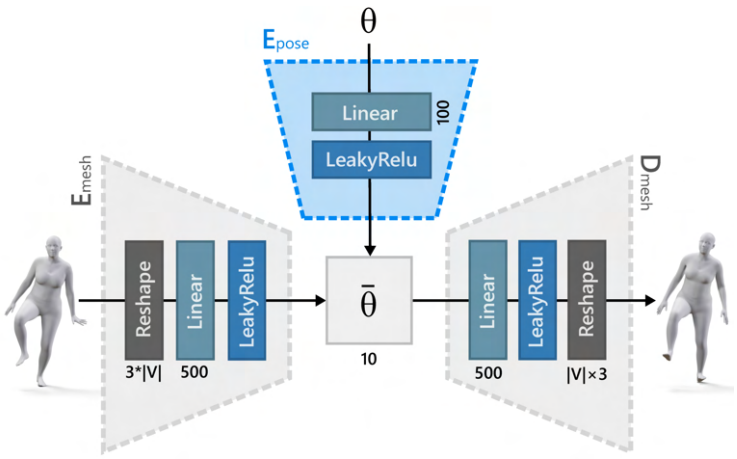


Figure 3.3: Architecture of the multi-modal pose autoencoder.

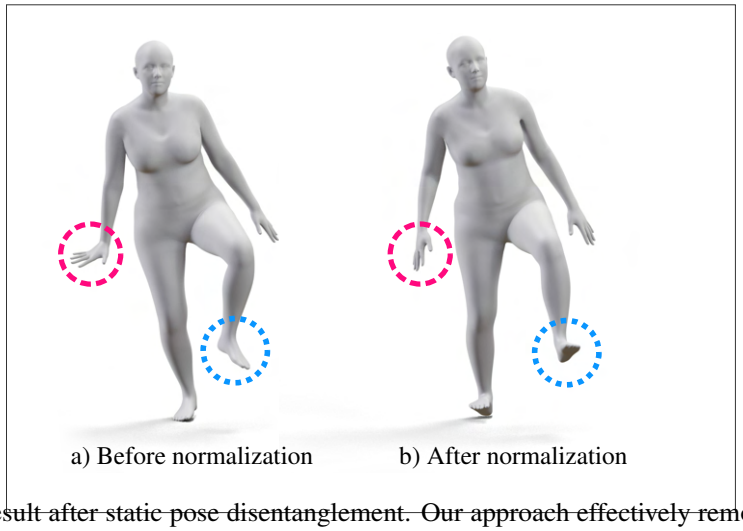
dataset of 4D scans, which requires a well disentangled and discriminative space to avoid overfitting tissue deformation and pose. Importantly, notice that removing these features manually is not feasible, not only because of the required time, but also because these features are not always apparent to a human observer. We therefore propose a novel and effective approach to *deshape* the pose coefficients, *i.e.*, to disentangle subject-specific anatomical features into a normalized and low-dimensional pose space $\bar{\theta}$:

$$\bar{\theta} = E_{\text{pose}}(\theta). \tag{3.12}$$

We find $E_{\text{pose}}(\theta) \in \mathbb{R}^{|\bar{\theta}|}$ by training a multi-modal encoder-decoder architecture, shown in Figure 3.3. In particular, having a mesh scan $\mathbf{S}$ and its corresponding pose θ and shape β parameters (found by solving Equation 3.4), we simultaneously train two encoders and one decoder minimizing the loss

$$\mathcal{L} = \|M(\theta, \mathbf{0}) - D_{\text{mesh}}(E_{\text{mesh}}(M(\theta, \mathbf{0}))\|_2 + \|M(\theta, \mathbf{0}) - D_{\text{mesh}}(E_{\text{pose}}(\theta)\|_2, \tag{3.13}$$

where $M(\theta, \mathbf{0})$ are the surface vertices of a skinned mesh in pose θ and mean shape (*i.e.*, vector of shape coefficients is zero). The intuition behind this multi-modal autoencoder is the following: the encoder E_{mesh} takes as input *skinned vertices* to enforce the similarity of large deformations (*e.g.*, lifting arms, where many vertices move) in the autoencoder loss. By using a significantly small latent space, we are able to simultaneously train it with the encoder E_{pose} such that the latter learns to remove undesired local pose articulations (and keep global deformations) directly in the pose vector θ. In contrast, notice that without the loss term that uses E_{mesh} we would not be able to distinguish between large and small deformations, because in the pose parameterization space of θ all parameters (*i.e.*, degrees of freedom) contribute equally.



a) Before normalization b) After normalization

Figure 3.4: Result after static pose disentanglement. Our approach effectively removes subject- and shape-dependent features, while retaining the main characteristics of the input pose.

The effect of the encoder E_{pose} is depicted in Figure 3.4, where subject- and shape-specific features are effectively removed, producing a *normalized* pose. In other words, we are disentangling features originally present in the pose descriptor θ (*e.g.*, wrist articulation) that are related to that particular subject or shape, but we are keeping the overall pose (*e.g.*, raising left leg). We found 10 to be an appropriate size of the latent space for a trade-off between capturing subtle motions and removing subject-specific features.

3.3.2 Avoiding dynamic pose entanglement

The novel pose representation $\bar{\theta}$ introduced earlier effectively disentangles *static* subject-specific features from the naive pose representation θ, however, our motion descriptor γ also takes temporal information (velocities and accelerations) into account. We observe that such temporal information can encode *dynamic* shape- and subject-specific features, causing an entanglement potentially making our regressor prone to overfitting soft-tissue deformations to subject-specific pose dynamics.

We address this by extending our 4D dataset by transferring sequences (encoded using our motion descriptor) across the different subjects. In particular, given two sequences of two different subjects

$$\mathcal{S}^i_{\text{A}} = \{\mathbf{S}^i_{\text{A},t}(\theta^i_t)\}_{t=0}^{N^i_A} \tag{3.14}$$

$$\mathcal{S}^j_{\text{B}} = \{\mathbf{S}^j_{\text{B},t}(\theta^j_t)\}_{t=0}^{N^j_B} \tag{3.15}$$

where $\mathbf{S}^i_{\mathrm{A},t}(\theta^i_t)$ is the mesh of the subject A performing the sequence identity i at time t, we transfer the sequence of poses θ^i_t to a subject B by training a subject-specific regressor R_B. This process generates a new sequence

$$\mathcal{S}^i_{\mathrm{B}} \quad = \quad R_{\mathrm{B}}(\gamma^{\,i}_{\mathrm{A}}) = \{\mathbf{S}^i_{\mathrm{B},t}(\theta^i_t)\}_{t=0}^{N^i_{\mathrm{A}}} \tag{3.16}$$

with the shape identity of the subject B performing the motion θ^i_t (notice, a motion originally performed by subject A). By transferring all motions across all characters, we are enriching our dataset in a way that effectively avoids overfitting soft-tissue deformations to subject and shape-specific dynamics (*i.e.*, style).

In Section 3.5 we detail the number of sequences and frames that we transfer, and evaluate the impact of this strategy. Specifically, Figure 3.7 shows an ablation study on how the generalization capabilities of our method improve when applying the pose disentangling methods introduced in this section.

3.4 Implementation details

In this section we provide details about the datasets, network architectures, and parameters to train our models.

3.4.1 Soft-tissue autoencoder and regressor

Data. Our soft-tissue autoencoder and soft-tissue regressor (Section 3.2.3) are trained using the 4D sequences provided in the Dyna dataset [Pon*15]. This dataset contains highly detailed deformations of registered meshes of 5 female subjects performing a total of 52 dynamic sequences captured at 60fps (42 used for training, 6 for testing). Notice that we do not use the Dyna provided meshes directly, but preprocess them to *unpose* the meshes. To this end, we solve Equation 3.4 for each mesh, and subsequently apply Equation 3.3 to find the ground truth displacements for all Dyna meshes.

Moreover, in addition to the motion transfer technique described in Section 3.3.2, we further synthetically augment the dataset by mirroring all the sequences.

Setup. We implement all networks in TensorFlow, including the encoder-decoder architecture of E_{soft} and D_{soft}, and the R regressor. We also leverage TensorFlow and its automatic differentiation capabilities to solve Equation 3.4. In particular, we optimize β using the first frame of a sequence and then optimize θ while leaving β constant. We use Adam optimizer with a learning rate of 1e-4 for the autoencoder and 1e-3 for the regressor. The autoencoder is trained during 1000 epochs (around 3 hours) with a batch size of 256, and a dropout rate of 0.1. The regressor is trained during 100 epochs (around 25 minutes) with batch size of 10 and no dropout. The details of the architecture are shown in Figure 3.2.

3.4.2 Pose autoencoder

Data. To train our pose autoencoder presented in Section 3.3.1 we are not restricted to the data of 4D scans because we do not need soft-tissue information. We therefore leverage the SURREAL dataset [Var*17], which contains a vast amount of Motion Capture (MoCap) sequences, from different actors, parameterized by pose representation θ. Our training data consists of 76094 poses from a total of 298 sequences and 56 different subjects, including the 5 subjects of the soft-tissue dataset (excluding the sequences used for testing the soft-tissue networks).

Setup. We use Adam optimizer with a learning rate of 1e-3, and a batch size of 256, during 20 epochs (20 min). The details of the architecture are shown in Figure 3.3.

3.5 Evaluation

In this section we provide qualitative and quantitative evaluation of both the reconstruction accuracy of our soft-tissue deformation subspace, described in Section 3.2.2, and the regressor proposed in Section 3.2.3.

3.5.1 Soft-tissue autoencoder evaluation

Quantitative evaluation. Table 3.1 shows a quantitative evaluation of the reconstruction accuracy of the proposed nonlinear autoencoder (AE) for soft-tissue deformation, for a variety of subspace sizes. We compare it with linear approaches based on PCA used in

previous works [Lop*15; Pon*15], in the full test dataset. These results demonstrate that our autoencoder consistently outperforms the reconstruction accuracy of the subspaces used in previous methods.

	25D	50D	100D
PCA	3.82mm	3.17mm	2.38mm
AE	**3.02mm**	**2.58mm**	**2.09mm**

Table 3.1: Reconstruction error of our soft-tissue autoencoder and PCA evaluated in the full test dataset. The autoencoder (AE) performs better than the linear approach (PCA) in all tested subspace sizes.

Qualitative evaluation. Figure 3.5 depicts a qualitative evaluation of the soft-tissue deformation autoencoder for a variety of subspace dimensions. Importantly, we also show that the reconstruction accuracy is attained across different shapes.

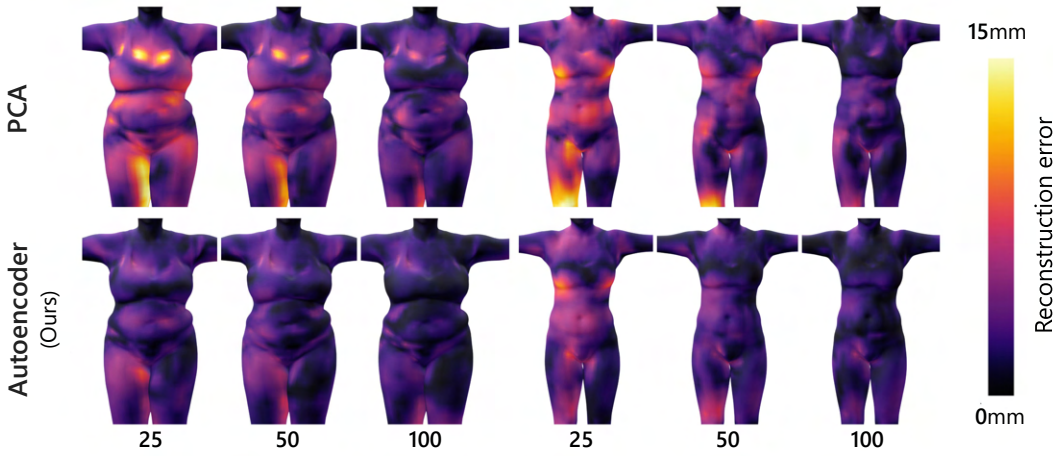


Figure 3.5: Reconstruction errors of our soft-tissue autoencoder and PCA, for two different body shapes. Notice that our subspace efficiently encodes soft-tissue displacements for parametric shapes, in contrast to previous works [CO18] that required an autoencoder per subject.

3.5.2 Soft-tissue regressor evaluation

We follow a similar evaluation protocol as in Dyna [Pon*15] and evaluate the following scenarios to exhaustively test our method. Additionally, we provide novel quantitative

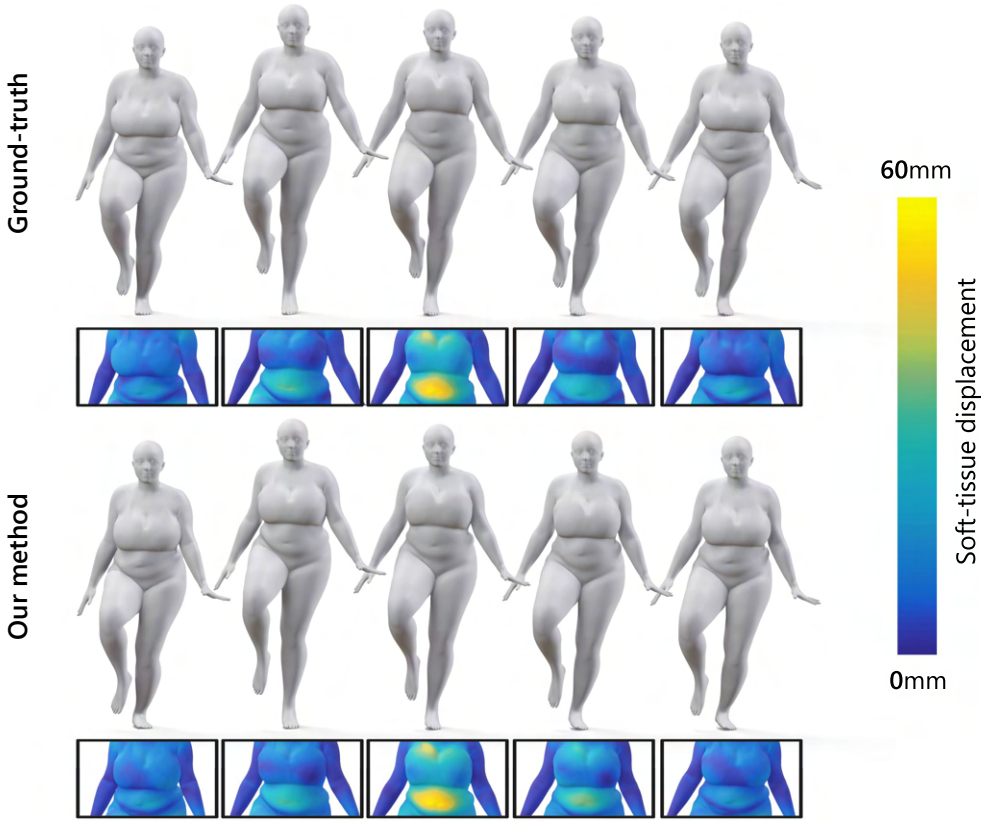


Figure 3.6: Evaluation of generalization to new motions. The sequence `one_leg_jump` was left out at train time, and used only for testing, for subject 50004. We show ground truth meshes and vertex displacements Δ^{GT} (top), and the regressed deformations Δ (bottom). Notice how the magnitude of the regressed displacement closely matches the ground truth.

insights that demonstrate significantly better generalization capabilities of our regression approach with respect to existing methods.

Generalization to new motions. In Figure 3.6 we demonstrate the generalization capabilities of our method to unseen motions. In particular, at train time, we left out the sequence `one_leg_jump` of the Dyna dataset and then used our regressor to predict soft-tissue displacements for this sequence, for the shape identity of the subject 50004. Leaving ground truth data out at train time allows us to quantitatively evaluate this scenario. To this end, we also show a visualization of the magnitude of soft-tissue displacement for both ground truth Δ^{GT} and regressed Δ displacements, and conclude that the regressed values closely match the ground truth.

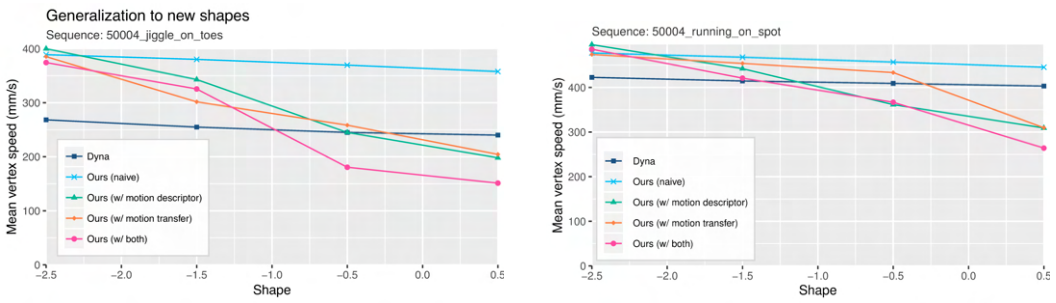


Figure 3.7: We quantitatively evaluate the generalization to new shapes of our regressor by looking at the mean vertex speed of the predicted soft-tissue offsets in unposed state in two test sequences. Our model (pink) produces a higher range of dynamics, with large velocities for obese subjects (shape parameter -2.5) and small velocities for thin subjects (shape parameter 0.5). In contrast, previous works (Dyna, in dark blue) produce a much smaller range, resulting in limited generalization capabilities to new subjects. Furthermore, here we also demonstrate that all components of our method contribute to getting the best generalization capabilities.

Generalization to new subjects. We quantitatively evaluate the generalization to new subjects by looking at the magnitude of the predicted soft-tissue displacements for different body shapes. Intuitively, subjects with larger body mass (which map to smaller $\beta[1]$ values) should exhibit larger soft-tissue velocities. In contrast, thin subjects (which map to mostly positive values in $\beta[1]$) should exhibit much lower soft-tissue velocities due to the high rigidity of their body surface. We exhaustively evaluate this metric in Figure 3.7, where we show an ablation study comparing our full method, our method trained with each of the contributions alone, and Dyna. Although Dyna [Pon*15] produces different deformation modes for different subjects, the resulting motion is significantly attenuated. In contrast, our full model (in pink) regresses a higher dynamic range of deformations, outputting larger deformations for small values of $\beta[1]$ (*i.e.*, obese subjects), and small surface velocities for larger values of $\beta[1]$ (*i.e.*, thin subjects). Importantly, we show that each contribution of our model (the static and dynamic pose disentangling methods introduced in Section 3.3) contributes to our final results, and that all together produce the highest range of deformations.

Generalization to new motion and new subject. We finally demonstrate the capabilities of our model to regress soft-tissue deformations for new body shapes and motions. To this end, we use MoCap data from SURREAL and AMASS datasets [Var*17; Mah*19] and arbitrary body shape parameters. Figure 3.8 shows sample frames of sequences `01_01` and `09_10` for two different shapes. Colormaps on 3D meshes depict per-vertex magnitude regressed offsets to reproduce soft-tissue dynamics. As expected, frames with more dynamics exhibit larger deformations.

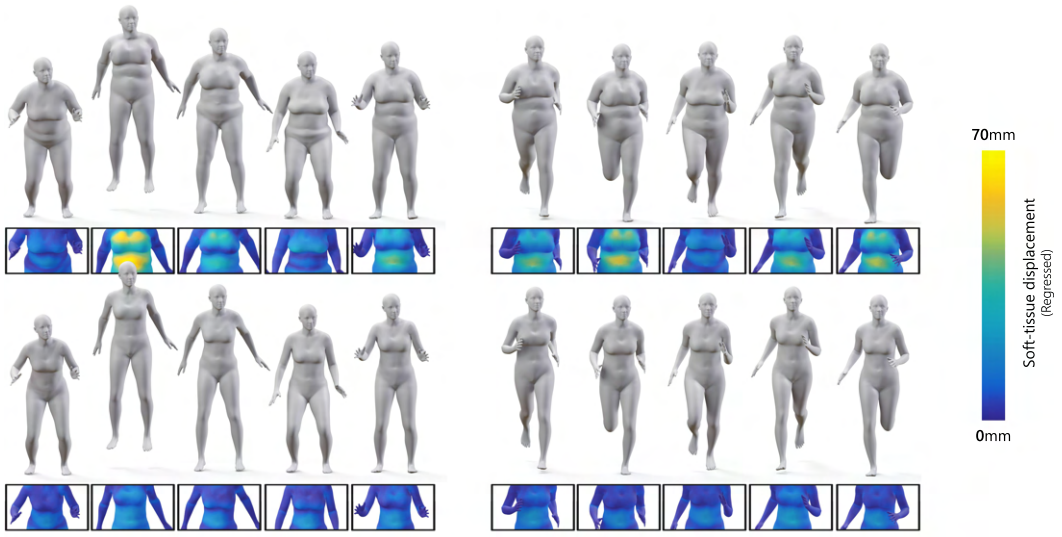


Figure 3.8: Sample frames of soft-tissue regression on two test sequences and two test subjects. Colormap depicts the magnitude of the regressed deformation. Notice how our method successfully regresses larger deformations on highly dynamic poses such as in the middle of a jump or when a foot steps on the ground.

3.5.3 Runtime performance

We have implemented our method on a regular desktop PC equipped with an AMD Ryzen 7 2700 CPU, a Nvidia GTX 1080 GPU, and 32GB of RAM. After training the model, we use TensorRT [NVI18] to optimize the neural networks for faster inference at runtime. On average, a forward pass of the optimized model takes 4.8ms. This cost is distributed across the components of the model as follows: 0.6ms the pose encoder, 1.9ms the soft-tissue regressor and 2.3ms the soft-tissue decoder.

3.6 Conclusions

In this chapter we have presented SoftSMPL, a data-driven method to model soft-tissue deformations of human bodies. Our method combines a novel motion descriptor and a recurrent regressor to generate per-vertex 3D displacements that reproduce highly expressive soft-tissue deformations. We have demonstrated that the generalization capabilities of our regressor to new shapes and motions significantly outperform existing methods. Key to our approach is to realize that traditional body pose representations rely on an entangled space that contains static and dynamic subject-specific features. By proposing a new disentangled

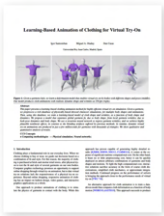
motion descriptor, and a novel subspace and regressor, we are able to model soft-tissue deformations as a function of body shape and pose with unprecedented detail.

Despite the significant step forward towards modeling soft-tissue dynamics from data, our method suffers for the following limitations. With the current 4D datasets available, which contain very few subjects and motions, it is not feasible to learn a model for a high-dimensional shape space. Furthermore, subtle motions that introduce large deformations are also very difficult to reproduce. Finally, as in most data-driven methods, our model cannot interact with external objects and does not support different topologies. Physics-based models can handle arbitrary meshes and react to external forces [Kim*17; Kad*16; KB18; Rom*20], but they come at the expense of significantly higher computational cost.

Our approach to static pose disentanglement depends on compression, which is not always reliable and requires choosing an appropriate size for the pose space. Since the dataset contains several subjects performing similar motions, future works could make use of this information to find more robust ways to disentangle pose from static subject features.

4 Supervised learning of garment deformations

This chapter presents a data-driven method to produce detailed clothing animations at interactive frame rates. In contrast to soft-tissue, clothing deformations exhibit higher frequency details (*e.g.*, folds and wrinkles) that change rapidly when the body moves and are harder to predict. To ease the learning task, our method is built upon standard skinning techniques, which we use to obtain an approximate model of the garment's motion. We then enhance this model by introducing a vector of corrective offsets that are computed by a recurrent neural network. In order to obtain realistic animations, the network learns these offsets from physically simulated sequences. We show that our method solves some of the visual limitations of previous works and is capable of generating plausible results in unseen motions and body shapes. Moreover, our method can be easily integrated into existing skeletal animation pipelines with little computational overhead. The contributions presented in this chapter have led to the following publication:



Igor Santesteban, Miguel A. Otaduy, and Dan Casas.
"Learning-Based Animation of Clothing for Virtual Try-On".
Computer Graphics Forum (Proc. Eurographics) (2019)

4.1 Introduction

Computer graphics technologies provide an opportunity to support online shopping through virtual try-on animation, but to date virtual try-on solutions lack the responsiveness required to provide an interactive and enjoyable experience. Beyond online shopping, responsive animation of clothing has an impact on fashion design, video games, and interactive graphics applications as a whole.

One approach to produce animations of clothing is to simulate the physics of garments in contact with the body. While this approach has proven capable of generating highly detailed

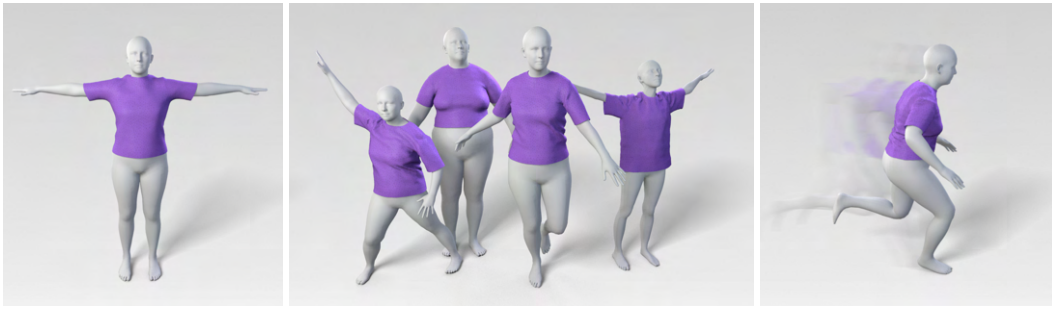


Figure 4.1: Given a garment (left), we learn a deformation model that enables virtual try-on by bodies with different shapes and poses (middle). Our model produces cloth animations with realistic dynamic drape and wrinkles at 250 fps (right).

results [KJM08; Sel*09; NSO12; Cir*14], it comes at the expense of significant runtime computational cost. On the other hand, it bears no or little preprocessing cost, hence it can be quickly deployed on almost arbitrary combinations of garments and body shapes and motions. To fight the high computational cost, interactive solutions sacrifice accuracy in the form of coarse cloth discretizations, simplified cloth mechanics, or approximate integration methods. Continued progress on the performance of solvers is bringing the approach closer to the performance needs of virtual try-on [Tan*18].

An alternative approach for cloth animation is to train a data-driven model that computes cloth deformation as a function of body motion [Wan*10; De *10]. This is similar to the approach we follow in the previous chapter to learn soft-tissue dynamics, but unlike soft-tissue, cloth exhibits rapidly changing folds and wrinkles that move and blend in a highly nonlinear manner. Data-driven methods succeed to produce plausible cloth deformations when there is a strong correlation between body pose and cloth deformation. However, early data-driven methods [Kav*11; Gua*12; Hah*14] struggle to represent the nonlinear behavior of cloth dynamics and contact in general. Most data-driven methods rely to a certain extent on linear techniques, hence the resulting wrinkles deform in a seemingly linear manner (*e.g.*, with blending artifacts) and therefore lack realism.

Most previous data-driven cloth animation methods work for a given garment-avatar pair, and are limited to representing the influence of body pose on cloth deformation. In virtual try-on, however, a garment may be worn by a diverse set of people, with corresponding avatar models covering a range of body shapes. In this chapter, we propose a learning-based method for cloth animation that meets the needs of virtual try-on, as it models the deformation of a given garment as a function of body motion *and* shape. Other methods that account for changes in body shape do not deform the garment in a realistic way, and either resize the garment while preserving its style [Gua*12; Bro*12], or retarget cloth wrinkles to bodies of different shapes [Pon*17; LCT18].

We propose a two-level strategy to learn the complex nonlinear deformations of clothing. On one hand, we learn a model of garment fit as a function of body shape. And on the other hand, we learn a model of local garment wrinkles as a function of body shape and motion. Our two-level strategy allows us to disentangle the different sources of cloth deformation.

We compute both the garment fit and the garment wrinkles using nonlinear regression models, *i.e.*, artificial neural networks, and hence we avoid the problems of linear data-driven models. Furthermore, we propose the use of recurrent neural networks to capture the dynamics of wrinkles. Thanks to this strategy, we avoid adding an external feedback loop to the network, which typically requires a dimensionality reduction step for efficiency reasons [CO18].

Our learning-based cloth animation method is formulated as a pose-space deformation, which can be easily integrated into skeletal animation pipelines with little computational overhead. We demonstrate example animations such as the ones in Figure 4.1, with a runtime cost of just 4ms per frame (more than $1000\times$ speed-up over a full simulation) for cloth meshes with thousands of triangles, including collision postprocessing.

To train our learning-based model, we leverage state-of-the-art physics-based cloth simulation techniques [NSO12], together with a parametric human model [Lop*15] and publicly available motion capture data [CMU; Var*17]. In addition to the cloth animation model, we have created a new large dataset of dressed human animations of varying shapes and motions.

4.2 Method

In this section, we describe our learning-based data-driven method to animate the clothing of a virtual character. Figure 4.2 shows an overview of the method, separating the preprocessing and runtime stages.

In Section 4.2.1 we overview the components of our shape-and-pose-dependent cloth deformation model. The two key novel ingredients of our model are: (i) a garment fit regressor (Section 4.2.2), which allows us to apply global body-shape-dependent deformations to the garment, and (ii) a garment wrinkle regressor (Section 4.2.3), which predicts dynamic wrinkle deformations as a function of body shape and pose.

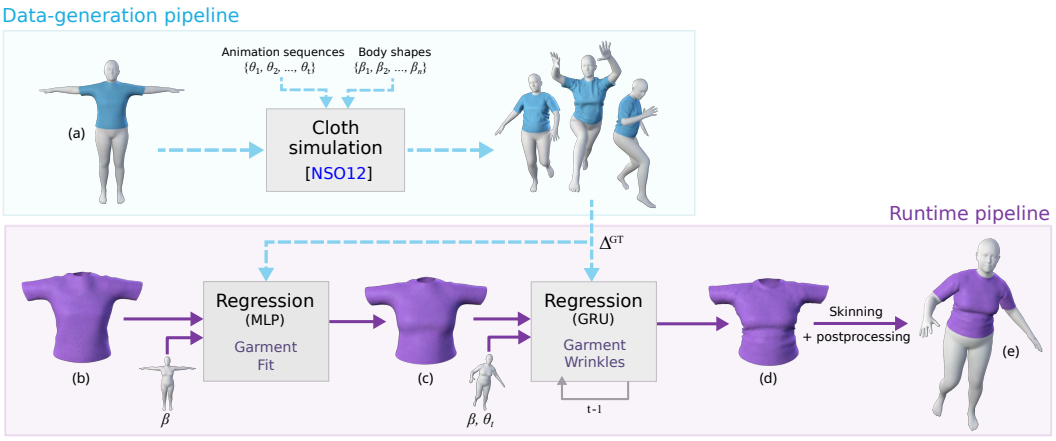


Figure 4.2: Overview of our preprocessing and runtime pipelines. As a preprocess, we generate physics-based simulations of multiple animated bodies wearing the same garment. At runtime, our data-driven cloth deformation model works by computing two corrective displacements on the unposed garment: global fit displacements dependent on the body's shape, and dynamic wrinkle displacements dependent on the body's shape and pose. Then, the deformed cloth is skinned on the body to produce the final result.

4.2.1 Clothing model

We denote as M_{b} a deformed human body mesh, determined by shape parameters β (*e.g.*, the principal components of a database of body scans) and pose parameters θ (*e.g.*, joint angles). We also denote as M_{c} a deformed garment mesh worn by the human body mesh. A physics-based simulation would produce a cloth mesh $S_{\text{c}}(\beta, \theta)$ as the result of simulating the deformation and contact mechanics of the garment on a body mesh with shape β and pose θ. Instead, we approximate S_{c} using a data-driven model.

Based on the observation that most garments closely follow the deformations of the body, we design our clothing model inspired by the Pose Space Deformation (PSD) literature [LCF00] and subsequent human body models [Ang*05; FCS15; Lop*15]. We assume that the body mesh is deformed according to a rigged parametric human body model,

$$M_{\text{b}}(\beta, \theta) = W(T_{\text{b}}(\beta, \theta), \beta, \theta, \mathcal{W}_{\text{b}}), \tag{4.1}$$

where $W(\cdot)$ is a skinning function, which deforms an unposed body mesh $T_{\text{b}}(\beta, \theta) \in \mathbb{R}^{3 \times V_{\text{b}}}$ with V_{b} vertices based on: first, the shape parameters $\beta \in \mathbb{R}^{|\beta|}$, which define joint locations of an underlying skeleton; and second, the pose parameters $\theta \in \mathbb{R}^{|\theta|}$, which are the joint angles to articulate the mesh according to a skinning weight matrix $\mathcal{W}_{\text{b}}$. The unposed body

mesh may be obtained additionally by deforming a template body mesh $\bar{\mathbf{T}}_{\mathrm{b}}$ to account for body shape and pose-based surface corrections (see, *e.g.*, [Lop*15]).

We propose to model cloth deformations following a similar overall pipeline. For a given garment, we start from a template cloth mesh $\bar{\mathbf{T}}_{\mathrm{c}} \in \mathbb{R}^{3\times V_{\mathrm{c}}}$ with V_{c} vertices, and we deform it in two steps. First, we compute an unposed cloth mesh $T_{\mathrm{c}}(\beta,\theta)$, and then we deform it using the skinning function $W(\cdot)$ to produce the full cloth deformation. A key insight in our model is to compute body-shape-dependent garment fit and shape-and-pose-dependent garment wrinkles as corrective displacements to the template cloth mesh, to produce the unposed cloth mesh:

$$T_c(\beta,\theta) = \bar{\mathbf{T}}_{\mathrm{c}} + R_{\mathrm{G}}(\beta) + R_{\mathrm{L}}(\beta,\theta), \tag{4.2}$$

where $R_{\mathrm{G}}()$ and $R_{\mathrm{L}}()$ represent two nonlinear regressors, which take as input body shape parameters and shape and pose parameters, respectively.

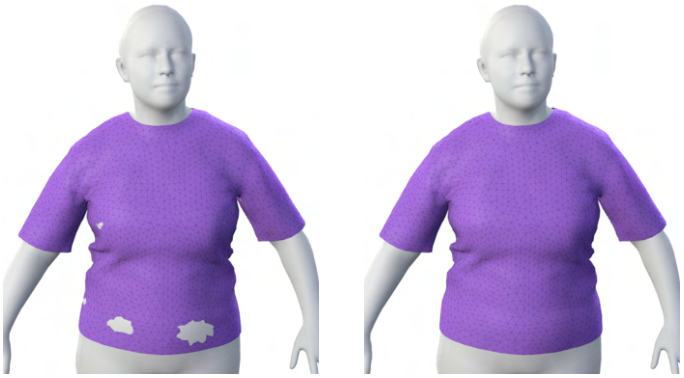
The final cloth skinning step can be formally expressed as

$$M_{\mathrm{c}}(\beta,\theta) = W(T_{\mathrm{c}}(\beta,\theta),\beta,\theta,\mathcal{W}_{\mathrm{c}}). \tag{4.3}$$

We define the skinning weight matrix $\mathcal{W}_{\mathrm{c}}$ by projecting each vertex of the template cloth mesh onto the closest triangle of the template body mesh, and interpolating the body skinning weights $\mathcal{W}_{\mathrm{b}}$.

The pipeline Figure 4.2 shows the template body mesh $\bar{\mathbf{T}}_{\mathrm{b}}$ wearing the template cloth mesh $\bar{\mathbf{T}}_{\mathrm{c}}$ (Figure 4.2-a), and then the template cloth mesh in isolation (Figure 4.2-b), with the addition of garment fit (Figure 4.2-c), with the addition of garment wrinkles (Figure 4.2-d), and the final deformation after the skinning step (Figure 4.2-e).

By training regressors with collision-free data, our data-driven model learns naturally to approximate contact interactions, but it does not guarantee collision-free cloth outputs. In particular, when the garments are tight, interpenetrations with the body can become apparent. After the skinning step, we apply a postprocessing step to cloth vertices that collide with the body, by pushing them outside their closest body primitive. An example of collision postprocessing is shown in Figure 4.3.



(a) Before **(b)** After

Figure 4.3: For tight clothing, data-driven cloth deformations may suffer from apparent collisions with the body (left). We apply a simple postprocessing step to push colliding cloth vertices outside the body (right).

4.2.2 Garment fit regressor

Our learning-based cloth deformation model represents corrective displacements on the unposed cloth state, as discussed above. We observe that such displacements are produced by two distinct sources. On one hand, the shape of the body produces an overall deformation in the form of stretch or relaxation, caused by tight or oversized garments, respectively. As we show in this section, we capture this deformation as a static global fit, determined by body shape alone.

On the other hand, body dynamics produce additional global deformation and small-scale wrinkles. We capture this deformation as time-dependent displacements, determined by both body shape and motion, as discussed later in Section 4.2.3. We reach higher accuracy by training garment fit and garment wrinkles separately, in particular due to their static vs. dynamic nature.

We characterize static garment fit as a vector of per-vertex displacements $\Delta_{\text{G}} \in \mathbb{R}^{3\times V_{\text{c}}}$. These displacements represent the deviation between the cloth template mesh $\bar{\mathbf{T}}_{\text{c}}$ and a smoothed version of the simulated cloth worn by the unposed body. Formally, we define the ground-truth garment fit displacements as

$$\Delta_{\text{G}}^{\text{GT}} = \rho(S_{\text{c}}(\beta, \mathbf{0})) - \bar{\mathbf{T}}_{\text{c}}, \tag{4.4}$$

where $S_{\text{c}}(\beta, \mathbf{0}))$ represents a simulation of the garment on a body with shape β and pose $\theta = \mathbf{0}$, and ρ represents a smoothing operator.

To compute garment fit displacements in our data-driven model, we use a nonlinear regressor $R_{\mathrm{G}} : \mathbb{R}^{|\beta|} \rightarrow \mathbb{R}^{3\times V_{\mathrm{c}}}$, which takes as input the shape of the body β. In particular, we implement the regressor $\Delta_{\mathrm{G}} = R_{\mathrm{G}}(\beta)$ using a single-hidden-layer multilayer perceptron (MLP) neural network. We train the MLP network by minimizing the mean squared error between predicted displacements Δ_{G} and ground-truth displacements $\Delta_{\mathrm{G}}^{\mathrm{GT}}$.

See Figure 4.4 for a visualization of garment fit regression. Notice how the original template mesh has globally deformed but lacks pose-dependent wrinkles.

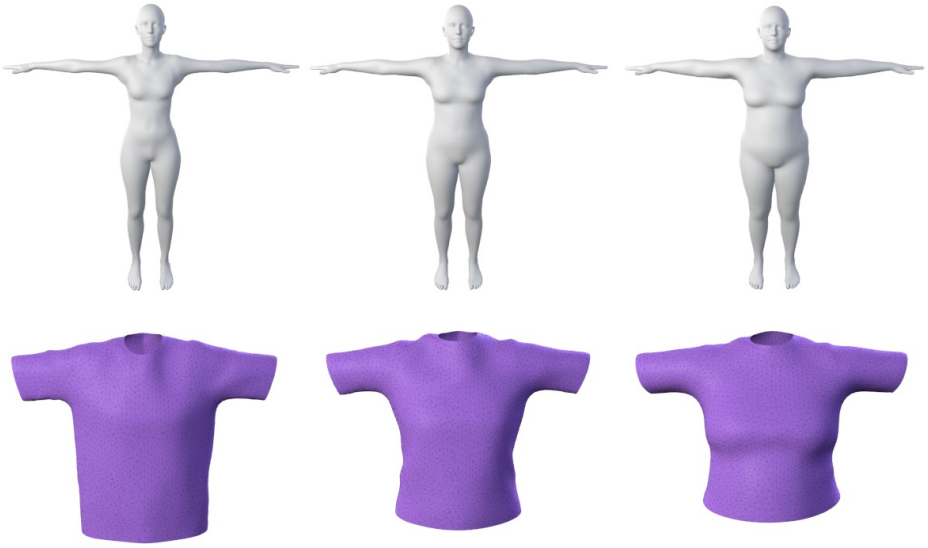


Figure 4.4: Results of garment fit regression for different bodies.

4.2.3 Garment wrinkle regressor

We characterize dynamic cloth deformations (*e.g.*, wrinkles) as a vector of per-vertex displacements $\Delta_{\mathrm{L}} \in \mathbb{R}^{3\times V_{\mathrm{c}}}$. These displacements represent the deviation between the simulated cloth worn by the moving body, $S_{\mathrm{c}}(\beta,\theta)$, and the template cloth mesh $\bar{\mathbf{T}}_{\mathrm{c}}$ corrected with the global garment fit Δ_{G}. We express this deviation in the body's rest pose, by applying the inverse skinning transformation $W^{-1}(\cdot)$ to the simulated cloth. Formally, we define the ground-truth garment wrinkle displacements as

$$\Delta_{\mathrm{L}}^{\mathrm{GT}} = W^{-1}(S_{\mathrm{c}}(\beta,\theta),\beta,\theta,\mathcal{W}_{\mathrm{c}}) - \bar{\mathbf{T}}_{\mathrm{c}} - \Delta_{\mathrm{G}}. \tag{4.5}$$

To compute garment wrinkle displacements in our data-driven model, we use a nonlinear regressor $R_{\mathrm{L}} : \mathbb{R}^{|\beta|+|\theta|} \rightarrow \mathbb{R}^{3\times V_{\mathrm{c}}}$, which takes as input the shape β and pose θ of the body. In contrast to the static garment fit, garment wrinkles exhibit dynamic, history-dependent deformations. We account for such dynamic effects by introducing recursion within the regressor. In particular, we implement the regressor $\Delta_{\mathrm{L}} = R_{\mathrm{L}}(\beta, \theta)$ using a Recurrent Neural Network (RNN) based on Gated Recurrent Units (GRU) [Cho*14], which has proven successful in modeling dynamic systems such as in human pose prediction [MBR17]. Importantly, GRU networks do not suffer from the well-known vanishing and exploding gradients common in vanilla RNNs [PMB13]. Analogous to the MLP network in the garment fit regressor, we train the GRU network by minimizing the mean squared error between predicted displacements Δ_{L} and ground-truth displacements $\Delta_{\mathrm{L}}^{\mathrm{GT}}$.

See Figure 4.5 for a visualization of garment wrinkle regression. Notice how the template garment fit to the body shape, obtained in the first step of our pipeline, is further deformed and enriched with pose-dependent dynamic wrinkles.

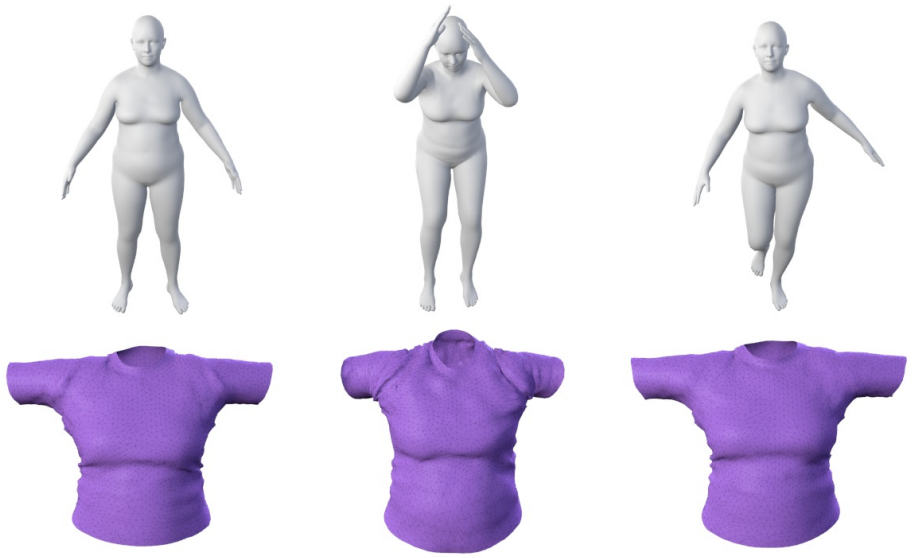


Figure 4.5: Results of garment wrinkle regression for different poses.

4.3 Implementation details

In this section, we give details on the generation of synthetic training sequences and the extraction of ground-truth data to train the neural networks. In addition, we discuss the network settings and the hyperparameters used in our results.

4.3.1 Dataset

To produce ground-truth data for the training of the regressors, we have created a novel dataset of dressed character animations with diverse motions and body shapes. Our prototype dataset has been created using only one garment, but it can be applied to other garments or their combinations. In Chapter 6 we use the same approach to create a similar dataset for a dress.

As explained in Section 4.2.1, our approach relies on the use of a parametric human model. In our implementation, we have used SMPL [Lop*15]. We have selected 17 training body shapes, as follows. For each of the 4 principal components of the shape parameters β, we generate 4 samples, leaving the rest of the parameters in β as 0. To these 16 body shapes, we add the nominal shape with $\beta = 0$.

As animations, we have selected character motions from the CMU dataset [CMU], applied to the SMPL body model [Var*17]. Specifically, we have used 56 sequences containing $7,117$ frames in total (at 30 fps, downsampled from the original CMU dataset of 120 fps). We have simulated each of the 56 sequences for each of the 17 body shapes, wearing the same garment mesh (*i.e.*, the T-shirt shown throughout the chapter, which consists of $8,710$ triangles).

All simulations have been produced using the ARCSim physics-based cloth simulation engine [NSO12; NPO13], with remeshing turned off to preserve the topology of the garment mesh. ARCSim requires setting several material parameters. In our case, since we are simulating a T-shirt, we have chosen an interlock knit with 60% cotton and 40% polyester, from a set of measured materials [WOR11]. We have executed all simulations using a fixed time step of 3.33ms, with the character animations running at 30 fps and interpolated to each time step. We have stored in the output database the simulation results from 1 out of every 10 time steps, to match the frame rate of the character animations. This produces a total of $120,989$ output frames of cloth deformation.

ARCSim requires a valid collision-free initial state. To this end, we manually pre-position the garment mesh *once* on the template body mesh $\bar{\mathbf{T}}_{\mathrm{b}}$. We run the simulation to let the cloth relax, and thus define the initial state for all subsequent simulations. In addition, we apply a smoothing operator $\rho(\cdot)$ to this initial state to obtain the template cloth mesh $\bar{\mathbf{T}}_{\mathrm{c}}$.

The generation of ground-truth garment fit data requires the simulation of the garment worn by unposed bodies of various shapes. We do this by incrementally interpolating the shape parameters from the template body mesh to the target shape, while simulating the

garment from its collision-free initial state. Once the body reaches its target shape, we let the cloth rest, and we compute the ground-truth garment fit displacements Δ_G^{GT} according to Equation 4.4.

Similarly, to simulate the garment on animations with arbitrary pose and shape, we incrementally interpolate both shape and pose parameters from the template body mesh to the shape and initial pose of the animation. Then, we let the cloth rest before starting the actual animation. The simulations produce cloth meshes $S_c(\beta, \theta)$, and from these we compute the ground-truth garment wrinkle displacements Δ_L^{GT} according to Equation 4.5.

4.3.2 Networks and training

We have implemented the neural networks presented in Sections 4.2.2 and 4.2.3 using TensorFlow [Mar*15]. The MLP network for garment fit regression contains a single hidden layer with 20 hidden neurons, which we found enough to predict the global fit of the garment. The GRU network for garment wrinkle regression also contains a single hidden layer, but in this case we obtained the best fit of the test data using 1500 hidden neurons. In both networks, we have applied dropout regularization to avoid overfitting the training data. Specifically, we randomly disable 20% of the hidden neurons on each optimization step. Moreover, we shuffle the training data at the beginning of each training epoch.

We have implemented the training process using the Adam optimization method [KB15], and we train our models for 2000 epochs with the learning rate set to 0.001. For the garment fit MLP network, we train using the ground-truth data from all 17 body shapes. For the garment wrinkle GRU network, we train using the data from 52 animation sequences, leaving 4 sequences for testing purposes. When training the GRU network, we use a batch size of 128 and use Truncated Backpropagation Through Time (TBPTT) with a limit of 90 frames, which reduces training times.

4.4 Evaluation

In this section, we discuss quantitative and qualitative evaluation of the results obtained with our method. We compare our results with other state-of-the-art methods, and we demonstrate the benefits of our method for virtual try-on, in terms of both visual fidelity and runtime performance.

4.4.1 Quantitative evaluation

Generalization to new body shapes. In Figure 4.6, we quantitatively evaluate the generalization of our method to new shapes (*i.e.*, not in the training set). We depict the per-vertex mean error on a static pose (left) and a dynamic sequence (right), as we change the body shape over time. To provide a quantitative comparison to existing methods, we additionally show the error suffered by cloth retargeting [LCT18; Pon*17]. Retargeting methods scale the garment in a way analogous to the body to retain the garment's style. Even if retargeting produces appealing results, it does not suit the purpose of virtual try-on, and produces larger error w.r.t. a physics-based simulation of the garment. This is clearly visible in Figure 4.6, where the error with retargeting increases as the shape deviates from the nominal shape, while it remains stable with our method.

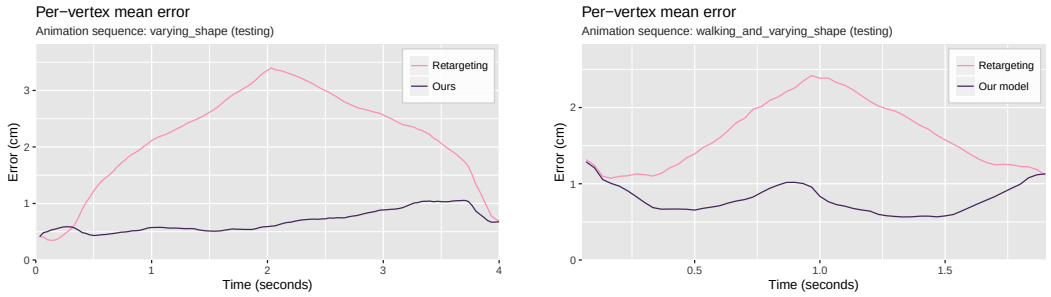


Figure 4.6: Quantitative evaluation of generalization to new shapes, comparing our method to retargeting techniques [LCT18; Pon*17]. The top plot shows the error as we increase the body shape to values not used for training, and back, on a static pose (see Figure 4.9). The bottom plot shows the error as we change both the body shape and pose during a test sequence not used for training.

Generalization to new body poses. In Figure 4.7, we depict the per-vertex mean error of our method in 2 test motion sequences with constant body shape but varying pose. In particular, we validate our cloth animation results on the CMU sequences 01_01 and 55_27 [CMU], which were excluded from the training set, and exhibit complex motions including jumping, dancing and highly dynamic arm motions. Additionally, we show the error suffered by two baseline methods for cloth animation. On one hand, Linear Blend Skinning (LBS), which consists of applying the kinematic transformations of the underlying skeleton directly to the garment template mesh. On the other hand, a Linear Regressor (LR) that predicts cloth deformation directly as a function of pose, implemented using a single-layer MLP. The results demonstrate that our two-step approach, with separate nonlinear regression of garment fit and garment wrinkles, outperforms the linear approach.

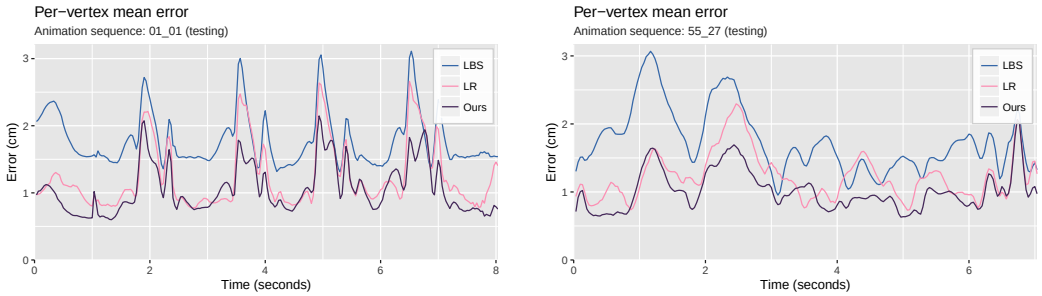


Figure 4.7: Quantitative evaluation of generalization to new poses, comparing our method to Linear Blend Skinning (LBS) and Linear Regression (LR).

Linear *vs.* nonlinear regression. In Figure 4.8, we compare the fitting quality of our nonlinear regression method *vs.* linear regression (implemented using a single-layer MLP), on a training sequence. While our method retains the rich and history-dependent wrinkles, linear regression suffers smoothing and blending artifacts.

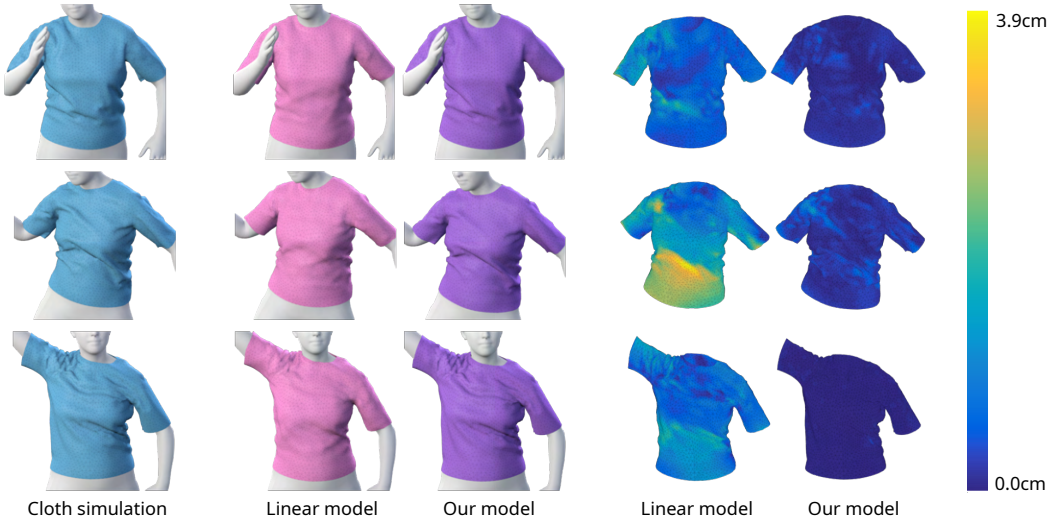


Figure 4.8: Our nonlinear regression method succeeds to retain the rich and history-dependent wrinkles of the physics-based simulation. Linear regression, on the other hand, suffers blending and smoothing artifacts even on the training sequence shown in the figure.

4.4.2 Qualitative evaluation

Generalization to new shapes. In Figure 4.9, we show the clothing deformations produced by our approach on a static pose while changing the body shape over time. We compare results with a physics-based simulation and with retargeting techniques [LCT18;

Pon*17]. Notice how our method successfully reproduces ground-truth deformations, including the overall drape (*i.e.*, how the T-shirt slides up the belly due to the stretch caused by the increasingly obese character) and mid-scale wrinkles.

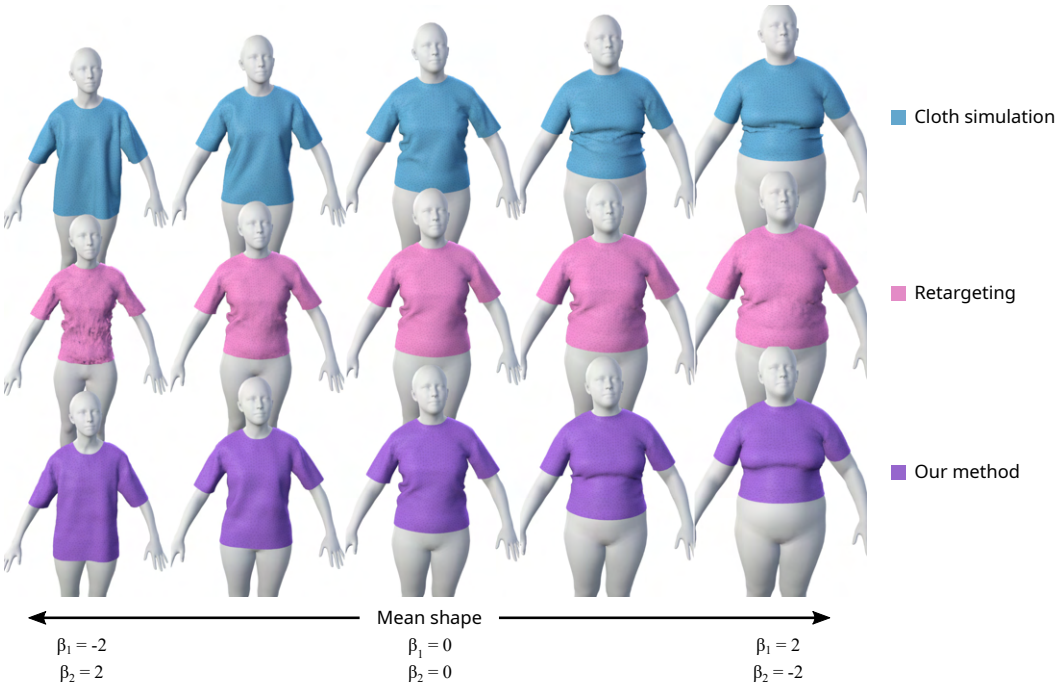


Figure 4.9: Our method matches qualitatively the deformations of the ground-truth physics-based simulation when changing the body shape beyond training values. In particular, notice how the T-shirt achieves the same overall drape and mid-scale wrinkles. Retargeting techniques [LCT18; Pon*17], on the other hand, scale the garment, and suffer noticeable artifacts away from the base shape.

We also compare our method to state-of-the-art data-driven methods that account for changes in both body shape and pose. Figure 4.10 shows the result of DRAPE [Gua*12] when the same garment is worn by two avatars with significantly different body shapes. DRAPE approximates the deformation of the garment by scaling it such that it fits the target shape, which produces plausible but unrealistic results. In contrast, our method deforms the garment in a realistic manner.

In Figure 4.11, we compare our model to ClothCap [Pon*17], a performance capture method that reconstructs clothing and shape independently, from 4D scans. A potential application of ClothCap is to retarget or transfer the captured garment to different shapes. However, retargeting lacks realism because cloth deformations are simply copied across different shapes. In contrast, our method produces realistic pose- *and* shape-dependent deformations.

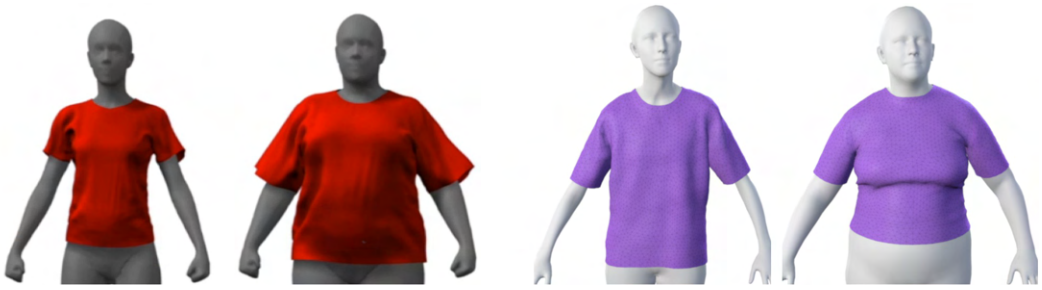


Figure 4.10: Comparison between DRAPE [Gua*12] (left) and our method (right). DRAPE cannot realistically cope with shape variations, and it is limited to scaling the garment to fit the target shape. In contrast, our method predicts realistically how a garment fits avatars with very diverse body shapes.

Figure 4.11: Comparison between ClothCap [Pon*17] (left) and our method (right). In ClothCap, the original T-shirt (top-left) is obtained using performance capture, and then scaled to fit a bigger avatar. While the result appears plausible for certain applications, it is not suited for virtual try-on. In contrast, our method produces pose- *and* shape-dependent drape and wrinkles, thus enabling a virtual try-on experience.

Generalization to new poses. We visually evaluate the quality of our model in Figure 4.13, where we compare ground-truth physics-based simulation and our data-driven cloth deformations on a test sequence. The overall fit and mid-scale wrinkles are successfully predicted using our data-driven model, with a performance gain of three orders of magnitude. Similarly, in Figure 4.12 we show more frames of a test sequence. Notice the realistic wrinkles in the belly area that appear when the avatar crouches.

Figure 4.12: Cloth animation produced by our data-driven method on a test sequence.

Figure 4.13: Comparison between a ground-truth physics-based simulation (top) and our data-driven method (bottom), on a test sequence not used for training (01_01 from [CMU]). Even though our method runs three orders of magnitude faster, it succeeds to predict the overall fit and mid-scale wrinkles of the garment.

4.4.3 Runtime performance

We have implemented our method on an Intel Core i7-6700 CPU, with a Nvidia Titan X GPU and 32GB of RAM. Table 4.1 shows average per-frame execution times of our implementation including garment fit regression, garment wrinkle regression, and skinning, with and without collision postprocessing. For reference, we also include simulation timings of a CPU-based implementation of full physics-based simulation using ARCSim.

The low computational cost of our method makes it suitable for interactive applications. Its memory footprint is as follows: 1.1MB for the garment fit regressor, and 108.1MB for the garment wrinkle regressor, both without any compression.

	ARCSim [NSO12]	Our method (w/o postprocess)	Our method (w/ postprocess)
mean	5635.4 ms	1.51 ms	4.01 ms
std	2488.5 ms	0.28 ms	0.27 ms

Table 4.1: Per-frame execution times of our method, with and without collision postprocessing. Full physics-based simulation times are also provided for reference.

4.5 Conclusions

In this chapter, we have presented a novel data-driven method for clothing animation that enables efficient virtual try-on applications at over 250 fps. Given a garment template worn by a human model, our two-level regression scheme independently models two distinct sources of deformation: garment fit, due to body shape; and garment wrinkles, due to shape and pose. We have shown that this strategy, in combination with the ability of the regressors to represent nonlinearities and dynamics, allows our method to overcome the limitations of previous data-driven approaches.

We believe our approach makes an important step towards bridging the gap between the accuracy and flexibility of physics-based simulation methods and the computational efficiency of data-driven methods. Nevertheless, there are a number of limitations that could be improved.

Our results show that our method succeeds at predicting the overall drape and mid-scale wrinkles of garments, but it smooths high-frequency wrinkles, both spatially and temporally. Additionally, our model is rooted on the assumption that most garments closely follow the body. This assumption may not be valid for loose clothing, where the decomposition of the deformation into a static fit and dynamic wrinkles would not yield accurate results.

Moreover, our method does not fully handle collisions between the body and the garment. Our regressors are trained with collision-free data, and therefore our model implicitly learns to approximate contact, but it is not guaranteed to be collision-free. Although a postprocess step can be effective at solving residual collisions, it adds a significant computational overhead and can yield unrealistic results when solving large collisions. Chapter 6 addresses this limitation by imposing low-level collision constraints as an explicit objective for the model.

Finally, our method requires independent training per garment. Given the low computational cost of the regressor, it would be possible to animate multiple garments at the same time, but each garment needs its own dataset and training, which takes hundreds of hours to compute. The next chapter presents an improved method that overcomes this limitation by removing the need of precomputing a dataset.

5 Self-supervised learning of garment deformations

This chapter presents a self-supervised method to learn dynamic 3D deformations of garments worn by parametric human bodies. Most data-driven approaches to model 3D garment deformations, including the method presented in the previous chapter, are trained using supervised strategies that require large datasets, usually obtained by expensive physics-based simulation methods or professional multi-camera capture setups. In contrast, this chapter proposes a new training scheme that removes the need for ground-truth samples, enabling self-supervised training of dynamic 3D garment deformations. Our key contribution is to realize that physics-based deformation models, traditionally solved in a frame-by-frame basis by implicit integrators, can be recast as an optimization problem. We leverage such optimization-based scheme to formulate a set of physics-based loss terms that can be used to train neural networks without precomputing ground-truth data. This allows us to learn models for interactive garments, including dynamic deformations and fine wrinkles, with a two orders of magnitude speed up in training time compared to state-of-the-art supervised methods. The contributions presented in this chapter have led to the following publication:

Igor Santesteban, Miguel A. Otaduy, and Dan Casas.
"SNUG: Self-Supervised Neural Dynamic Garments".
Proc. of Computer Vision and Pattern Recognition (CVPR) (2022)

5.1 Introduction

The efficient modeling of digital garments is an active area of research due to the large number of applications, including fashion design, e-commerce, virtual try-on, and video games. The traditional approach to this problem is through physics-based simulation [Nea*06], but the high computational cost required at runtime hinders the deployment of these techniques to real-world applications. Learning-based approaches such as the method presented in Chapter 4 and the models proposed by other works [PLP20; Gun*19; Ma*20; Vid*20;

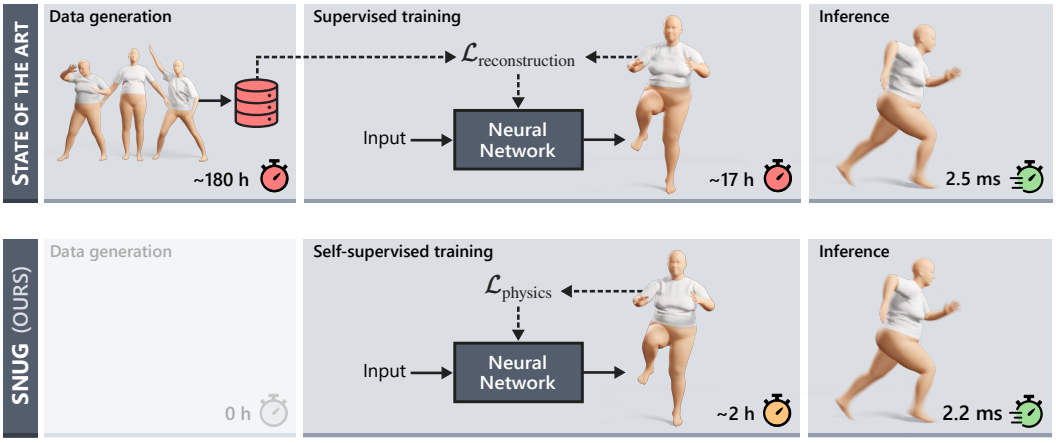


Figure 5.1: Existing learning-based methods for garment deformations (top) use supervised training schemes that require the expensive computation of large datasets. In contrast, our approach SNUG (bottom) is a learning-based method that enables the self-supervised training of dynamic neural 3D garments, without requiring any ground-truth data.

Tiw*20; Wan*18; SLL20] have demonstrated that it is possible to closely approximate the accuracy of physics-based solutions. These methods use *supervised* learning strategies to find a function that outputs a deformed garment given an input body descriptor. During the training phase, the supervision is enforced by directly minimizing at a vertex level the difference between the predicted garment and ground-truth 3D meshes. Despite requiring hours of training, learning-based methods are highly-efficient to evaluate at runtime, therefore they potentially offer an attractive alternative to traditional physics-based solutions.

However, the need for large datasets in current supervised methods is far from ideal. Ground-truth meshes must be obtained –for each combination of garment, body shape, and pose– via computationally-expensive simulations [NSO12] or complex 3D scanning setups [Pon*17], which heavily hinders the scalability of current learning-based methods. We observe that for similar image-based problems, *self-supervised* strategies have shown that it is possible to learn complex tasks without requiring ground-truth data [Raj*18; Wu*19]. Unfortunately, self-supervision for dynamic 3D clothing has not been explored.

In this work, we present a self-supervised method to learn dynamic deformations of 3D garments worn by parametric human bodies. The key to our success is realizing that the solution to the equations of motion used in current physics-based methods can also be formulated as an optimization problem [Mar*11]. More specifically, we show that the per-time-step numerical integration scheme used to update the vertex position (*e.g.*, backward Euler) in physics-based simulators, can be recast as an optimization problem, and demonstrate that the function for this minimization can become the central ingredient of a self-supervised learning scheme. Since this objective function includes both an inertial

term and static term directly derived from the equations of motion, we are able to learn time-dependent and pose-dependent deformations *without* any ground-truth data.

The advantages of self-supervision go beyond removing the need for ground-truth data. By reformulating the learning task in terms of physics-based intrinsic properties instead of explicit 3D surface similarity, we also mitigate the smoothing artifacts common in supervised methods where L2 losses are used directly at the vertex level [PLP20]. Additionally, self-supervised approaches also generalize better to test sequences outside the distribution of the training set. Finally, we also show how different material models can be easily formulated in our self-supervised framework, bringing the generalization capabilities of physics-based solutions (*i.e.*, deform any material) to learning-based methods, without requiring any precalculation or offline step.

All in all, our main contribution is a novel learning-based method capable of learning to dynamically deform garments using a self-supervised strategy. We demonstrate the superiority of our approach in terms of data requirements, training time, and inference time, and we quantitatively and qualitatively compare our results with state-of-the-art supervised methods.

5.2 Method

Our goal is to find a function $M()$ that deforms a 3D garment given the underlying body parameters and motion. To this end, in Section 5.2.1 we first describe our garment model used to implement $M()$, which is based on per-vertex dynamic 3D displacements that are added to a rigged template mesh. Then, in Section 5.2.2, we direct our attention to an optimization-based formulation of dynamic deformations. Based on this formulation, in Section 5.2.3, we introduce our main contribution and describe a physics-based deformation model that allows us to train a regressor $R()$ for 3D garment displacements. Importantly, our loss is driven by fundamental physical properties of deformable objects, not by the reconstruction of ground-truth garments, and therefore it enables *self-supervised* learning. In Section 5.2.4 we specify the material model used in the different terms of our loss, and define the relevant energies such as the strain, and bending energies. Finally, in Section 5.2.5 we describe the recurrent architecture used to implement the regressor $R()$. See Figure 5.2 for an overview of our method.

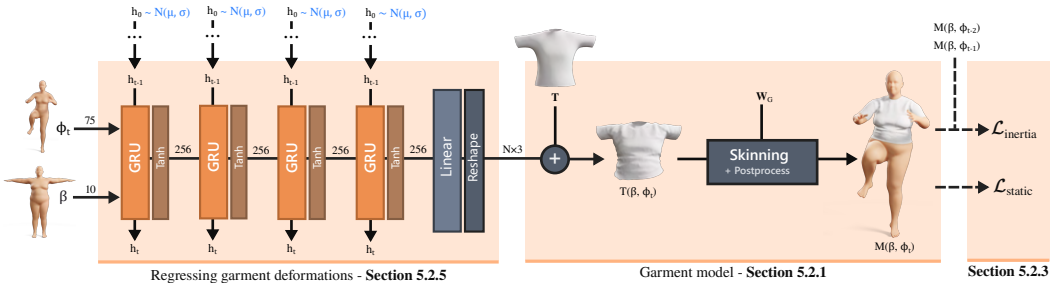


Figure 5.2: Overview of our method. First, the recurrent regressor predicts per-vertex offsets as a function of body shape and motion. These offsets are added to the garment template which is then skinned to produce the final result. We train the network by optimizing a set of physical properties of the predicted garments, removing the need for ground-truth data.

5.2.1 Garment model

Similar to the previous chapter as well as other state-of-the-art methods [Gun*19; PLP20; BME20; Vid*20], we leverage and extend existing human body models [Lop*15; FCS15] to encode garment deformations. More specifically, we build our representation on top of the popular SMPL human model [Lop*15]. SMPL encodes bodies by deforming a rigged human template according to shape and pose-dependent deformations that are learned from data. Following this idea, we define our garment model as

$$M(\beta, \phi) = W(T(\beta, \phi), J(\beta), \theta, \mathbf{W}_G) \tag{5.1}$$

$$T(\beta, \phi) = \mathbf{T} + R(\beta, \phi) \tag{5.2}$$

where W is a skinning function (*e.g.*, linear blend skinning or dual quaternion) with skinning weights $\mathbf{W}_G$, joint locations $J(\beta)$, and motion parameters ϕ that articulate an unposed deformed garment mesh $T(\beta, \phi)$. The latter is computed from a garment template mesh $\mathbf{T}$ deformed by a function $R(\beta, \phi)$ that outputs per-vertex 3D displacements to encode dynamic deformations conditioned to the underlying body shape β and body motion ϕ. The body motion ϕ contains the current body pose θ as well as the global velocity of the root joint.

Assuming that the garment template $\mathbf{T}$ is correctly located on top of the mean SMPL body mesh [Lop*15], we define $\mathbf{W}_G$ by borrowing the SMPL skinning weights of the closest body vertices in rest pose. In the remainder of this section we introduce our novel strategy to learn the 3D displacement regressor $R(\beta, \phi)$.

5.2.2 Optimization-based dynamic deformation

Our goal is to learn the 3D displacement regressor $R(\beta, \phi)$ in Equation 5.2 using a self-supervised strategy. To this end, our first task is to find a set of physics-based properties that describe how cloth behaves. Physics-based simulators traditionally solve dynamics by applying a numerical integration scheme, *e.g.*, backward Euler, to the differential equations of motion, and finding the roots of the resulting nonlinear discrete equations [Nea*06]. This formulation is applied independently at each simulation frame, to iteratively update the positions and velocities of garment vertices. Our key observation is to realize that the solution to the equations of motion discretized with backward Euler can also be formulated as an optimization problem [Mar*11], and the objective function for this minimization can become the central ingredient of a self-supervised learning scheme. Optimization-based dynamics have been used in the Computer Graphics literature to increase the efficiency and robustness of dynamics solvers, through quasi-Newton schemes and step-size selection [Gas*15; LBK17]. Instead, we propose to leverage such optimization-based formulation to define a loss for training a neural network that generalizes well to any input (*i.e.,* any body shape and motion).

The equations of motion can be discretized with backward Euler as

$$\mathbf{M}\frac{\mathbf{x}^{t+1} - \mathbf{x}^t - \Delta t \mathbf{v}^t}{\Delta t^2} = \mathbf{f}\left(\mathbf{x}^{t+1}, \frac{\mathbf{x}^{t+1} - \mathbf{x}^t}{\Delta t}\right), \tag{5.3}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{f}$ are forces, and $\mathbf{x}$ and $\mathbf{v}$ are the positions and velocities of garment nodes. The solution to these equations can be recast as an optimization [Mar*11; Gas*15]:

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} \frac{1}{2\Delta t^2}(\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{M}(\mathbf{x} - \hat{\mathbf{x}}) + \mathbf{\Phi}, \tag{5.4}$$

where $\hat{\mathbf{x}} = \mathbf{x}^t + \Delta t \mathbf{v}^t$ is a tentative (explicit) position update, and $\mathbf{\Phi}$ is the potential energy due to internal and external forces $\mathbf{f}$ of the system.

5.2.3 Turning dynamics into self-supervision

The key to our method is to define a set of losses based on Equation 5.4 to train the regressor $R()$. To this end, we propose a loss with two terms

$$\mathcal{L} = \mathcal{L}_{\text{inertia}} + \mathcal{L}_{\text{static}}, \tag{5.5}$$

where $\mathcal{L}_{\text{inertia}}$ models the inertia of the garment and it is defined analogous to the first term of Equation 5.4

$$\mathcal{L}_{\text{inertia}} = \frac{1}{2\Delta t^2}(\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{M}(\mathbf{x} - \hat{\mathbf{x}}). \tag{5.6}$$

Intuitively, this term prevents the change of garment velocities over time, but garment velocities will change anyway due to the underlying body motion, which makes dynamics and wrinkle effects appear.

$\mathcal{L}_{\text{static}}$, the second term of our loss $\mathcal{L}$, models the potential energy $\mathbf{\Phi}$ of Equation 5.4 which represents the internal and external forces that affect the garment. Inspired by works from cloth simulation literature [NSO12; SB12], we define $\mathcal{L}_{\text{static}}$ as the sum of different physics-based terms that model the energies that emerge on deformable solids, including strain, bending, gravity, and collisions

$$\mathcal{L}_{\text{static}} = \mathcal{L}_{\text{strain}} + \mathcal{L}_{\text{bending}} + \mathcal{L}_{\text{gravity}} + \mathcal{L}_{\text{collision}}. \tag{5.7}$$

This formulation of $\mathcal{L}_{\text{static}}$ is general, and the definition of each term depends on the material model used, which we detail in the next section.

5.2.4 Material model

The literature of simulation of elastic solids characterizes materials using equations that relate stimuli (*e.g.*, deformations) to material response (*e.g.*, energies) [SB12]. Inspired by this, and with the goal of learning physically-correct garment behaviors, we define the terms of our static loss $\mathcal{L}_{\text{static}}$ based on equations of state-of-the-art cloth simulators [NSO12] to model the following energies:

Membrane strain energy. The membrane strain term models the response of the material to in-plane deformation. Given a deformed position $x \in \mathbb{R}^3$ and an undeformed position $X \in \mathbb{R}^2$ (*i.e.*, the garment template), it defines an internal energy based on a first-order deformation metric, typically the deformation gradient $\mathbf{F} = \frac{\partial x}{\partial X}$. In our loss we implement it using the Saint Venant Kirchhoff (StVK) elastic material model that defines membrane strain energy as

$$\Psi_{\text{S}} = \frac{\lambda}{2}\text{tr}(\mathbf{G})^2 + \mu\text{tr}(\mathbf{G}^2), \tag{5.8}$$

where λ and μ are the Lamé constants, and $\mathbf{G} = \frac{1}{2}(\mathbf{F}^\top\mathbf{F} - \mathbf{I})$ is the Green strain tensor.

The membrane strain energy of the mesh is computed as

$$\mathcal{L}_{\text{strain}} = \sum_{\text{triangles}} \mathbf{V}\Psi_{\text{S}}, \tag{5.9}$$

where $\mathbf{V}$ is the volume of each triangle (*i.e.,* area $\times$ thickness).

Bending Energy. The bending term models the energy due to the angle of two adjacent faces and we model it as

$$\mathcal{L}_{\text{bending}} = \sum_{\text{edges}} k_{\text{bending}}\, k_{\text{scale}} \frac{\theta^2}{2} \tag{5.10}$$

where θ is the dihedral angle between the adjacent faces, k_{bending} is a bending stiffness, and $k_{\text{scale}} = l^2/4(a_1 + a_2)$ is a scaling factor that accounts for the area of the faces a_1, a_2 as well as the length l of the shared edge.

Gravity. To model the effect of gravity in the learned deformations, we add a loss term with the potential energy of each cloth vertex

$$\mathcal{L}_{\text{gravity}} = \sum_{\text{vertices}} -m\, \mathbf{g}^\top x \tag{5.11}$$

where m is the vertex mass, and $\mathbf{g}$ is the gravitational acceleration.

Collision penalty. This term is crucial to learn plausible deformations, enforcing the garment to follow the underlying body motion. We implement it as

$$\mathcal{L}_{\text{collision}} = \sum_{\text{vertices}} k_{\text{collision}}\, \max(\epsilon - d(x), 0)^3 \tag{5.12}$$

where $d(x)$ is a function that computes the signed distance to the body, $k_{\text{collision}}$ is a collision stiffness, and ϵ is a safety margin to prevent the garment from overlapping with the body surface.

To highlight the realism of the proposed material, in Figure 5.3 we show a ground-truth simulation of our model, and the simpler material model used in PBNS [BME21] based on a traditional mass-spring formulation. Overall, our model is capable of reproducing more complex behaviors typically present in garments, including wrinkles and folds at different scales.

(a) StVK (Ours) **(b)** Mass-spring (PBNS [BME21])

Figure 5.3: The material model used is crucial to obtain realistic garment behaviors. We formulate our losses using the Saint Venant Kirchoff (StVK) model, in contrast to simpler alternatives that lead to less expressive deformations.

5.2.5 Regressing garment deformations

With our novel self-supervised loss $\mathcal{L}$ defined in Section 5.2.3, we are ready to train the garment displacement regressor $R()$ from Equation 5.2 without requiring ground-truth data. To this end, in order to model the time dependencies of the inertial term $\mathcal{L}_{\text{inertia}}$, we implement the regressor using 4 Gated Recurrent Units (GRU), each with an output of size 256, and `tanh` as the activation function (see Figure 5.2). However, the recurrent nature of GRUs combined with the lack of ground-truth values to guide the training process make the regressor converge to bad solutions if a naive recurrent training protocol is used. We need to take special care into how the hidden states of the GRUs are initialized and updated.

Intuitively, the model should be able to learn dynamics from just 3 frames, since $\mathcal{L}_{\text{inertia}}$ from Equation 5.6 depends only on the vertex positions and velocities of the previous step. Therefore, we train our network using sub-sequences of 3 frames. Interestingly, we found that training on longer sub-sequences also minimizes $\mathcal{L}_{\text{inertia}}$ correctly, but the learned deformations do not model true dynamics.

At runtime, the network supports sequences of arbitrary length, but results can degrade noticeably for sequences longer than those used in training if initialization of the GRU hidden states is not well handled. More specifically, we observe that for each training sub-sequence, setting the initial hidden states $\mathbf{h}_0 = 0$ hinders the network to generalize to sequences longer than 3 frames. We address this issue by sampling the initial state $\mathbf{h}_0$ of each GRU from $\mathcal{N}(\mu, \sigma)$ (empirically, $\mu = 0$ and $\sigma = 0.1$), which allows the model to generalize well even for sequences with thousands of frames. Notice that at runtime

the state $\mathbf{h}_t$ depends on an arbitrarily large number of previous frames, not just the last 3, hence the use of noise to initialize states on train sub-sequences is fundamental to augment variance in states.

5.3 Evaluation

5.3.1 Training

To self-supervise the training process of our regressor $R()$ we need to feed it with human motions and shapes. To this end, we use a set of 52 sequences from the AMASS dataset [Mah*19], totaling 6,519 frames, which we split into sub-sequences of 3 frames as described in Section 5.2.5. We set aside 4 full sequences for validation purposes. To provide body shape variety at train time, each of the sub-sequences is assigned a different body shape β sampled from $\mathcal{U}(-3, 3)$ at each epoch. Notice that, enabled by our self-supervised approach, this strategy allows us to train using thousands of different body shapes, while competitive supervised methods are limited to a dramatically smaller shape sample (TailorNet [PLP20] uses 9 shapes, the method presented in Chapter 4 uses 17) due to the computational restrictions caused by the need for a ground-truth database.

Regarding the network hyper-parameters, we use a batch size of 16, initially train for 10 epochs using a learning rate of 0.001, and then resume the learning with a learning rate of 0.0001 until it converges. This approach is fast, works for all garments, and avoids erroneous states. The rest of the material and training parameters do not affect stability. Larger learning rates can introduce instabilities due to energy spikes that make the training struggle to recover (*i.e.*, the predicted mesh has collisions that are too large to be resolved). Small body-garment collisions are not a problem – *e.g.*, we can handle pants despite self-collisions in the legs on some poses.

Our approach does not require balancing loss terms, we just need to set the material properties of the garment. To this end, we tune material parameters to produce a desired fabric behavior, hence the parameters of the loss have a physical meaning – they are not arbitrary hyperparameters. To compute the mass matrix $\mathbf{M}$ we use real measurements of the thickness and density of 100% cotton fabric (0.47 mm and 426 kg/m^3 respectively). The rest of the material parameters have the following values: the Lamé constants are set to $\lambda = 4.44\text{e}4$ and $\mu = 2.36\text{e}4$, the bending stiffness $k_{\text{bending}} = 3.96\text{e}{-5}$, the collision

stiffness $k_{\text{collision}} = 250$, and the collision margin $\epsilon = 2$ mm. We use the same parameters for all our garments.

To thoroughly validate our model, in addition to comparisons to SOTA methods, in this section we also include ablations and comparisons that use a ground-truth simulated dataset. For as fair as possible evaluations, such dataset is created using the same motions and the same train-test split that we use to train SNUG.

We implement our method in a regular desktop PC equipped with an AMD Ryzen 7 2700 CPU, an Nvidia GTX 1080 Ti GPU, and 32GB of RAM.

5.3.2 Quantitative evaluation

To quantitatively evaluate our approach, we measure the physics-based terms of our loss $\mathcal{L}$ in test motions and compare it with the predictions of PBNS [BME21]. Notice that the original PBNS method uses a different (and simpler) material model but, in order to get a meaningful quantitative comparison, we extended and re-trained the publicly available PBNS implementation with our material model defined in Section 5.2.4. Also, notice that we cannot provide this comparison for supervised state-of-the-art methods (*e.g.*, [PLP20; SOC19; Gun*19]) because the simulation schemes, material models, and parameters used to build their datasets are different and, therefore, the ground-truth physics properties (*i.e.*, our loss terms) might differ significantly.

Figure 5.4 shows the quantitative evaluation for the most important terms of our loss, and compares it with the extended implementation of PBNS [BME21] using our material, in the test sequence `01_01` of AMASS [Mah*19].

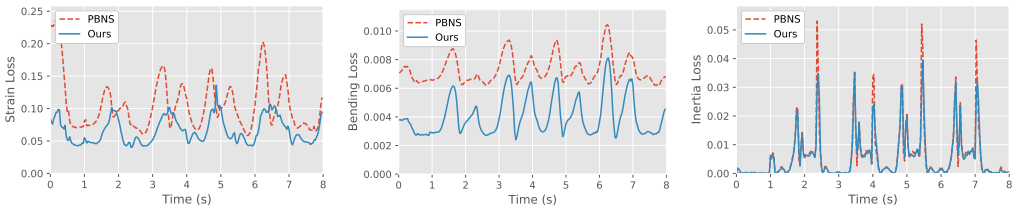


Figure 5.4: Quantitative evaluation of our approach. We evaluate the error in the physics-based terms used in our loss, in the test sequence `01_01` of AMASS [Mah*19]. Sudden motion changes (*e.g.*, jumps) naturally produce peaks in the inertial term, due to drastic changes in the velocity of the garment. Intuitively, cloth dynamics arise when the garment resists those changes induced by the body, therefore lower inertial values indicate that our model learns time-dependent effects better than PBNS [BME21].

Notice how our method consistently produces lower error values across all terms (strain, bending, and inertia), indicating that test samples processed with SNUG better match the behavior of physics-based solutions (*i.e.*, the minimization of the terms). Table 5.1 presents a quantitative evaluation of both methods in our full test set (4 sequences, 598 frames unseen at train time), which further demonstrates that our approach improves upon the method of PBNS.

	Strain	Bending	Gravity	Inertia
PBNS [BME21]	0.111	0.007	0.044	0.0035
SNUG (Ours)	**0.064**	**0.004**	**0.028**	**0.0034**

Table 5.1: To quantitative evaluate our method we compute the physics-based loss terms of our trained model, in unseen sequences, and compare to PBNS. We produce lower errors in all terms, indicating that our approach results in deformations that better match physics-based simulators.

To validate each term of our formulation, in Table 5.2 we show an ablation study of the mean-curvature error, evaluated in the test set of our ground-truth simulated dataset, when leaving out some of the terms.

	W/o bending	W/o strain	W/o gravity	W/o inertia	Full
Mean-curvature error	17.3	19.1	7.5	2.8	**2.7**

Table 5.2: Quantitative ablation study. Each term of our loss contributes to the accuracy of the final result.

Finally, in Table 5.3 we also evaluate the memory requirements, training time, and runtime performance of our approach and compare to existing state-of-the-art supervised methods. Even if these methods do no address exactly the same problem (*e.g.*, TailorNet [PLP20] models garment variations and SNUG does not, but the latter models dynamics), SNUG outperforms supervised methods by a large margin in all metrics, resulting in a compact model, only 19MB, trained in just 2h, which opens the door to scalable learning-based garment models.

	Data generation	Train	Runtime	Memory
TailorNet [PLP20]	29 h	6.5 h	10.1 ms	2114 MB
Chapter 4 [SOC19]	180 h	17 h	2.5 ms	109 MB
Chapter 5 (SNUG)	**0 h**	**2 h**	**2.2 ms**	**19 MB**

Table 5.3: Timings, memory requirements, and performance of state-of-the-art methods. Our self-supervised approach avoids the expensive cost of data generation, while also achieving significantly lower training times.

5.3.3 Qualitative evaluation

We qualitatively evaluate our method in Figure 5.5. To this end, notice that we always use body shapes and motions unseen during training. Additionally, we provide comparisons to the state-of-the-art *supervised* methods of Santesteban *et al.* [SOC19] (presented in Chapter 4) and TailorNet [PLP20], as well as to the recent work PBNS [BME21] that uses physics-constraints as supervision. To ease the assessment of the realism of each method, we also show results computed with a physics-based simulator [NSO12], but notice that this is a traditional offline method, several orders of magnitude slower.

These results demonstrate that our self-supervised method SNUG produces garment deformations that are, at least, on par with the state-of-the-art *supervised* methods [SOC19; PLP20], while we do not require *any* ground-truth dataset. For PBNS [BME21], we use a mean body shape because it does not generalize to different bodies. Because PBNS does not model an inertial term and it is limited to a simpler material model, the garment deformations are generally more stiff, less realistic, and do not change naturally as a function of body pose. This is visible in rows 1 and 3 for PBNS in Figure 5.5, where the overall wrinkles are the same despite the significant change in body pose.

To further validate our model, we use the ground-truth simulated dataset (described in Section 5.3.1, used for validation purposes only) to retrain our neural network in a per-vertex supervised manner. In Figure 5.6 we qualitatively demonstrate that the self-supervised method learns more detailed wrinkles than the supervised counterpart trained with exactly the same motions.

Additionally, in Figure 5.7 we show more results for a variety of garments learned with our approach, including T-shirts, tops, sleeveless shirts, pants, and shorts, worn by different body shapes. Notice how our approach produces different wrinkles for each garment type, pose,

and shape combination, demonstrating the generalization capabilities of our self-supervised approach. For this figure, we trained one regressor for each garment type.

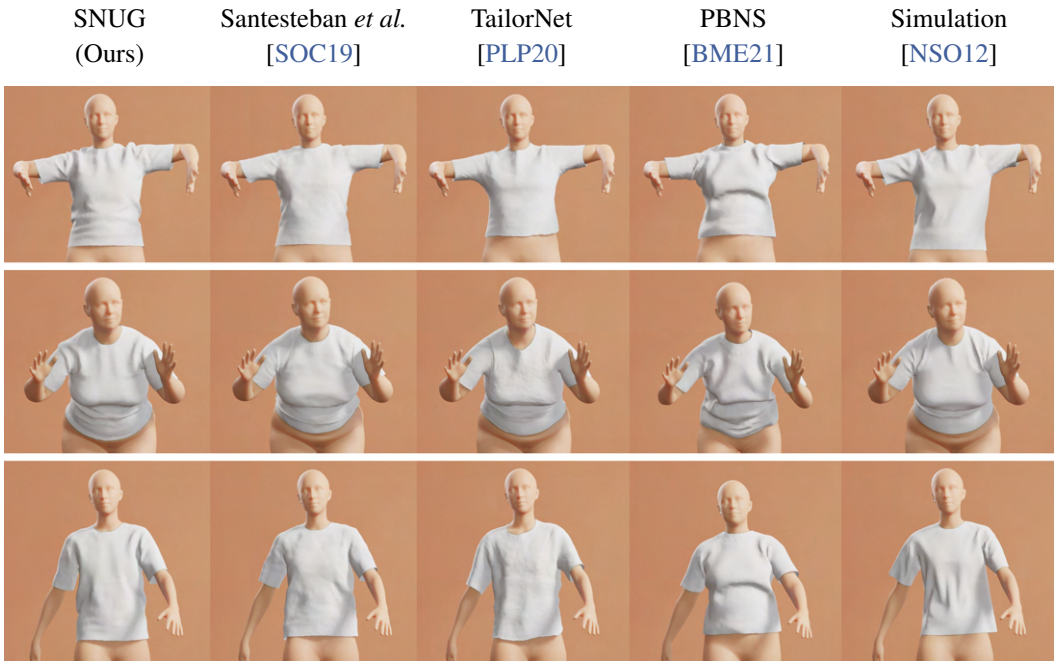


Figure 5.5: Qualitative comparison with state-of-the-art methods. SNUG generalizes well to unseen body shapes and motions and produces detailed folds and wrinkles. The results of SNUG are on par with the realism of *supervised* methods that require large datasets [SOC19; PLP20] and close to *offline physics-based* simulation [NSO12].

(c) Simulation

Figure 5.6: When trained using same motions and same architecture, direct supervision at the vertex level leads to smoothing artifacts (a). In contrast, our physics-based loss is able to learn more realistic details (b), as shown in this frame from a test sequence.

Figure 5.7: Qualitative results of our self-supervised method, in validation body shapes and poses unseen during training. SNUG successfully learns highly-realistic garment deformations, including fine wrinkles, as a function of body shape and motion.

5.4 Conclusions

We believe SNUG makes an important step towards efficient learning-based models for 3D garments. To improve the state-of-the-art, instead of following the standard route of training with more data, adding more explicit supervision, or designing more complex architectures, we show that self-supervision based on physical properties of deformable solids leads to simpler and smaller yet highly-realistic models.

While our physics-based loss terms are the fundamental key to self-supervision, we also want to point out that our strategy of exploiting optimization-based schemes (originally derived for simulation problems) to train a neural network carries a few weaknesses and important considerations to take into account.

Specifically, we notice that the self-supervised network tends to converge to simpler solutions than a traditional simulator. For example, although our approach is capable of learning pose- and shape-dependent wrinkles and overall dynamics, we struggle to predict fine-level dynamics. We hypothesize that this limitation arises from a fundamental difference in how our method works: while standard simulators solve physics for one frame at a time, our model optimizes thousands of frames simultaneously during training. This makes our approach more prone to converge to simpler local minima. Nevertheless, we want to highlight that, despite this limitation, the cloth dynamics learned by our method are on par with other data-driven approaches.

Another aspect open to improvement is the collision handling. Although our loss penalizes collisions between the garment and the body in train samples, we found noticeable collisions in test motions. Although these collisions can be efficiently solved with a postprocessing step, in the following chapter we explore alternative ways to enforce this constraint directly on the learned models.

6 Handling collisions between garments and bodies

In this chapter, we propose a new generative model for 3D garment deformations that enables us to learn, for the first time, a data-driven method for virtual try-on that effectively addresses garment-body collisions. In contrast to the methods presented in Chapters 4 and 5 that require an undesirable postprocessing step to fix garment-body interpenetrations at test time, this chapter presents a novel method that directly outputs 3D garment configurations that do not collide with the underlying body. Key to our success is a new canonical space for garments that removes pose-and-shape deformations already captured by a new diffused human body model, which extrapolates body surface properties such as skinning weights and blendshapes to any 3D point. We leverage this representation to train a generative model with a novel self-supervised collision term that learns to reliably solve garment-body interpenetrations. We extensively evaluate and compare our results with recently proposed data-driven methods, and show that our method is the first to successfully address garment-body contact in unseen body shapes and motions, without compromising realism and detail. The contributions presented in this chapter have led to the following publication:

Igor Santesteban, Nils Thuerey, Miguel A. Otaduy, and Dan Casas. "Self-supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On". *Proc. of Computer Vision and Pattern Recognition (CVPR)* (2021)

6.1 Introduction

The digitalization of 3D garments has important applications in many areas of our everyday lives such as online shopping, video games, visual effects, and fashion design, and it has traditionally been addressed with physics-based methods [NSO12; Nea*06]. However, even if these methods offer solutions that generalize well to any type of garment, produce physically-accurate results, and solve body-garment contact, they require computationally expensive runtime evaluations. Consequently, they do not meet the combined robustness

Figure 6.1: Our data-driven method regresses deformed garments via a generative model that is trained to avoid collisions.

and performance needed for real-time applications such as virtual try-on. Furthermore, they are not easily differentiable and cannot be integrated into computer vision pipelines that, for example, fit deformable models into images to extract information about the scene.

Data-driven methods have emerged as a popular alternative to physics-based methods. The core idea is to *learn* a function that mimics the garment behavior observed in a large dataset. To this end, recent methods leverage the capability of neural networks to learn nonlinear functions, and propose differentiable models that output 3D deformed garments as a function of the target shape, motion, style, size, and other design parameters [Vid*20; SOC19; PLP20; Tiw*20; Wan*18; Gun*19; Ma*20]. These methods showcase great realism and robustness, however, we identify a fundamental limitation in all existing works: despite using a loss term that penalizes unphysical body-garment interpenetrations at training time [Gun*19; BME20], predicted garments commonly suffer from body-garment interpenetrations in test sequences. The methods presented in Chapters 4 and 5 as well as other state-of-the-art methods [PLP20] address this problem with a postprocessing step that pushes the problematic regions of the garment, identified by exhaustive search, outside of the body.

The undesired interpenetrations arise from natural residual errors in test samples when optimizing neural networks which, combined with the extremely narrow gap between body surface and garment, can produce artifacts even if the predicted 3D mesh closely matches the ground truth deformed garment. In this chapter, we address this inherent limitation and propose, to the best of our knowledge, the first data-driven method to reliably solve garment-body interpenetrations *without* requiring any postprocessing step. We achieve this through three main contributions.

First, we propose to enhance existing human body models [Lop*15] by learning to smoothly expand the surface parameters to any 3D point. Intuitively, this allows us to model the deformation at any 3D point, *e.g.*, a vertex of a deformed loose garment, leveraging the deformation capabilities of existing human body models. This expanded human body represents a fundamental building block for our method.

Our second contribution addresses the common assumption, made by existing data-driven models, that garment deformations closely follow the underlying body deformations. This popular simplification is often used to define garment models that use skinning parameters based on the closest body vertex in *rest pose*, and subsequently articulate the garment using a standard linear blend skinning (LBS) approach. We show that simplified transformations to bring ground truth data into a normalized representation, *e.g.* via inverse LBS [SOC19; PLP20], cannot correctly represent the complex deformations that garments exhibit, and often introduce undesirable artifacts. Instead, we propose a garment model that represents deformations in a novel *unposed* and *deshaped* canonical space by removing deformations already captured via our expanded human body model. Since it yields correct skinning attributes for any 3D point, our garment model is designed to not to introduce collisions during projection operations between the canonical space and the posed space.

Our third and most important contribution is to leverage the novel canonical representation of garments to learn a generative subspace of deformations. Garments in this canonical space are encoded with respect to a *constant* reference body configuration. This not only gives an improved representation of garment deformations, but also allows us to reliably learn to solve collisions via self-supervision, by exhaustively sampling the generative space. We then learn a regressor that outputs deformations encoded in this subspace, and use our garment model to project them to the final deformed state. Since both the deformation subspace and the projection step are designed to avoid collisions, our final 3D garments do not interpenetrate the underlying body mesh, regardless the shape and pose parameter. In this chapter, we follow a supervised approach to learn garment deformations (*i.e.*, we use a dataset of simulated garments to train our models, as in Chapter 4), but the contributions related to collision handling are also applicable in self-supervised contexts (Chapter 5).

6.2 Overview

Our goal is to learn a function to predict how a 3D garment dynamically deforms given a target human body pose and shape. In contrast to existing methods [Gun*19; PLP20; SOC19; BME20], we put special emphasis in learning a model that directly outputs garment geometry that does not interpenetrate with the underlying human body, *i.e.,* it is physically correct after inference without requiring any post-processing. Hence, the final state is not compromised in terms of the regressed garment details such as wrinkles and dynamics.

To this end, in Section 6.3.1 we introduce an extension of standard statistical human body models [Lop*15] that learns to smoothly diffuse skinning surface parameters, such as rigging

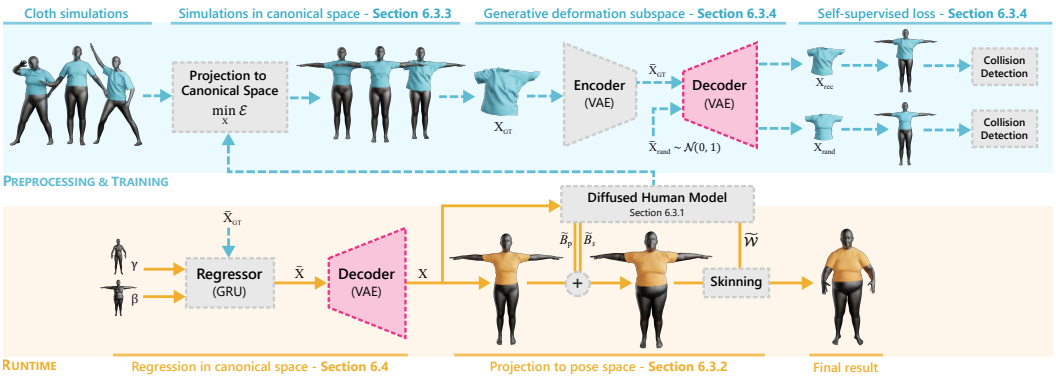


Figure 6.2: Overview of our preprocessing (top) and runtime pipelines (bottom). The decoder network is trained to avoid collisions in a self-supervised fashion, and then employed by the regressor network to reproduce these states at runtime.

weights and blendshape correctives, to any point in 3D space. In Section 6.3.2, we leverage these learned diffused skinning parameters to define a novel garment deformation model. The key idea is to remove the deformations already captured by our diffused body model to built an *unposed* and *deshaped* canonical space of garments. In this space, garments appear in rest pose and mean shape but pose- and shape-dependent wrinkle details are preserved. In Section 6.3.3, we introduce a novel optimization-based strategy to project physics-based simulations to our canonical space. Importantly, we show that the use of the learned diffuse skinning parameters is fundamental for this task, since they enable the correct representation of complex phenomena such as garment-body sliding or loose clothing.

Using projected physics-based simulations as ground-truth data, in Section 6.3.4 we describe how we learn a generative space of garment deformations. Key to our success is a novel self-supervised loss enabled by the canonical space of garments, which allows us to exhaustively sample *random* instances of garment deformations (*i.e.*, arbitrary shape, pose, and dynamics for which ground truth data is unavailable) and test collisions against a *constant* body mesh. Finally, in Section 6.4 we describe a recurrent regressor that outputs deformed garments with dynamics, that do not interpenetrate the body, as a function of body shape and motion.

6.3 Canonical space of garment deformations

The central aim of our method is to obtain a regressor R that infers the deformation of the garment via

$$\mathbf{X} = R(\beta, \gamma), \tag{6.1}$$

where $\mathbf{X} \in \mathbb{R}^{N_G \times 3}$ is the garment deformation in canonical space computed as a function of body shape β and motion descriptor γ. We will first describe how to obtain the canonical space into which the garment data is transformed, before detailing how the regressor R is trained in Section 6.4.

6.3.1 Diffused human model

Our garment model, defined later in Section 6.3.2, is driven by a new *diffused* human body model that extends current approaches in order to generalize to vertices beyond the body surface. More specifically, current body models [FCS15; Lop*15; JSS18] deform a rigged parametric human template

$$M_B(\beta, \theta) = W(T_b(\beta, \theta), J(\beta), \theta, \mathcal{W}) \tag{6.2}$$

where W is a skinning function (*e.g.*, linear blend skinning, or dual quaternion) with skinning weights $\mathcal{W}$ and pose parameters θ that deforms an unposed parametric body mesh $T_b(\beta, \theta)$. The nowadays standard SMPL model [Lop*15] defines the unposed body mesh as

$$T_B(\beta, \theta) = \mathbf{T}_b + B_s(\beta) + B_p(\theta) \tag{6.3}$$

where $\mathbf{T}_b \in \mathbb{R}^{N_B \times 3}$ is a body mesh template with N_B vertices that is deformed using two blendshapes that output *per-vertex* 3D displacements: $B_s(\beta) \in \mathbb{R}^{N_B \times 3}$ models deformations to change the body shape; and $B_p(\theta) \in \mathbb{R}^{N_B \times 3}$ models deformations to correct skinning artifacts. Follow-up works propose additional blendshapes to model soft-tissue [Pon*15; San*20] and garments [Ma*20; All*19; Pon*17].

The garment models from Chapters 4, 5 and existing works [PLP20] leverage the human body model defined in Equation 6.2 assuming that clothing closely follows the deformations of the body. Consequently, a common approach is to borrow the skinning weights $\mathcal{W}$ to model the articulation of garments, usually by exhaustively searching the nearest body vertex for each garment vertex in rest pose. Our key observation is that such naive static assignment cannot correctly model complex nonrigid clothing effects. The reason is twofold: first, the garment-body nearest vertex assignment must be dynamically updated, for example, when a garment slides over the skin surface; and second, the garment-body vertex assignment cannot be driven only by the closest vertex since this causes undesirable discontinuities in medial-axis areas.

To address these weaknesses, we propose to extend existing body models formulated in Equation 6.2 by smoothly *diffusing* skinning parameters to any 3D point around the body. It is worth mentioning that we are not the first to diffuse surface parameters, but previous works are limited to interpolate inwards to create a volumetric mesh [Kim*17; Rom*20] in a less smooth strategy. In Section 6.3.2, we show how our generalization of skinning parameters beyond the body surface is a fundamental piece for our novel garment model.

More formally, we define the functions $\widetilde{\mathcal{W}}(\mathbf{p})$, $\widetilde{B}_{\mathrm{s}}(\mathbf{p}, \theta)$, and $\widetilde{B}_{\mathrm{p}}(\mathbf{p}, \theta)$ that generalize skinning weights, shape blendshape offset, and pose blendshape offset, respectively, to any point $\mathbf{p} \in \mathbb{R}^3$ by smoothly diffusing the surface values

$$\widetilde{\mathcal{W}}(\mathbf{p}) = \frac{1}{N} \sum_{\mathbf{q}_n \sim \mathcal{N}(\mathbf{p}, \mathbf{d})} \mathcal{W}(\phi(\mathbf{q}_n)) \tag{6.4}$$

$$\widetilde{B}_{\mathrm{s}}(\mathbf{p}, \beta) = \frac{1}{N} \sum_{\mathbf{q}_n \sim \mathcal{N}(\mathbf{p}, \mathbf{d})} B_{\mathrm{s}}(\phi(\mathbf{q}_n), \beta) \tag{6.5}$$

$$\widetilde{B}_{\mathrm{p}}(\mathbf{p}, \theta) = \frac{1}{N} \sum_{\mathbf{q}_n \sim \mathcal{N}(\mathbf{p}, \mathbf{d})} B_{\mathrm{p}}(\phi(\mathbf{q}_n), \theta) \tag{6.6}$$

where $\phi(\mathbf{p})$ computes the closest surface point to $\mathbf{p} \in \mathbb{R}^3$, $\mathbf{d}$ is the distance from $\mathbf{p}$ to the surface body, and $B_{\mathrm{p}}(\mathbf{p}, \theta)$ is a function that returns the 3D offset of the vertex $\mathbf{p}$ computed by the blendshape B_{p}. Notice that, for each point, we average the values of N neighbors and therefore mitigate potential discontinuities in areas around a medial-axis.

In order to obtain differentiable functions that seamlessly integrate into an optimization or learning process, we employ recent works on learning neural fields [Par*19; Sit*20; Xie*22] and learn $\widetilde{\mathcal{W}}(\mathbf{p})$, $\widetilde{B}_{\mathrm{s}}(\mathbf{p}, \beta)$, and $\widetilde{B}_{\mathrm{p}}(\mathbf{p}, \theta)$ with fully-connected neural networks. This additionally yields a very efficient evaluation on modern GPUs.

6.3.2 Garment model

Our next goal is to define a garment model that is capable of representing the deformations naturally present in real garments, including dynamics, high-frequency wrinkles, and garment-skin sliding. To this end, a common approach to ease this task is to decouple the deformations caused by different sources, and model each case independently. For example, the method presented in Chapter 4 decouples deformations due to shape and pose, and Patel *et al.* [PLP20] due to shape, pose, and style. More specifically, as discussed in Section 6.3.1, both works model pose-dependent deformations leveraging the skinning weights associated with the body in the unposed state and a linear blend skinning technique. This

disentanglement removes many nonlinear deformations and enables to efficiently represent (and learn) deformations due to other sources directly in an *unpose* (*i.e.*, normalized) state.

We propose going one step further and removing the shape-dependent deformations already captured by the human body model. This effectively constructs a canonical *unposed* and *deshaped* representation of garments, improving the disentanglement proposed by earlier works. As we show later in Section 6.3.4, this is a fundamental step towards learning a generative space of garment deformations that do not interpenetrate the underlying body.

To formulate our unposed and deshaped garment model we leverage the diffused skinning functions proposed in Section 6.3.1

$$M_{\mathrm{G}}(\mathbf{X}, \beta, \theta) = W(T_{\mathrm{G}}(\mathbf{X}, \beta, \theta), J(\beta), \theta, \widetilde{\mathcal{W}}(\mathbf{X})) \tag{6.7}$$

$$T_{\mathrm{G}}(\mathbf{X}, \beta, \theta) = \mathbf{X} + \widetilde{B}_{\mathrm{s}}(\mathbf{X}, \beta) + \widetilde{B}_{\mathrm{p}}(\mathbf{X}, \theta), \tag{6.8}$$

where $T_{\mathrm{G}}()$ is the deformed garment after diffused blendshapes correctives are applied, and $\mathbf{X}$ are the garment deformations in canonical space. Notice that our garment model is well-defined for any garment with any topology, thanks to the generalized diffused skinning functions (*i.e.,* no need to retrain $\widetilde{\mathcal{W}}(), \widetilde{B}_{\mathrm{s}}(), \widetilde{B}_{\mathrm{p}}()$ for each garment).

The key property of this model is that skinning parameters used to articulate the garment (Equations 6.7 and 6.8) are defined as a function of the unposed and deshaped *deformed* garment $\mathbf{X}$. This is in contrast to existing methods [SOC19; PLP20] that use a *fixed* weight assignment, usually defined in a relaxed state or template, and cannot guarantee that the rigging step of the regressed deformed garment does not introduce collisions.

6.3.3 Projecting the ground-truth data

Our ultimate goal is to learn the function $R()$ from Equation 6.1, which predicts garment deformations in canonical space, in a data-driven manner. However, obtaining ground-truth data is not trivial since we need to project deformed 3D garments –computed with a physics-based simulator [NSO12]– to the unposed and deshaped space. Previous methods formulate the projection to the unposed state as the inverse of the linear blend skinning operation [PLP20; SOC19; Pon*17]. Due to their static rigging weights assignment, this operation can introduce body-garment collisions in the unposed state for frames where the garment has deformed significantly or slid in the tangential direction of the body (see Figure 6.3b). Even if a data-driven method can potentially learn to fix these artifacts to output collision-free *posed* deformations, our key contribution discussed in detail in Section 6.3.4 is to show

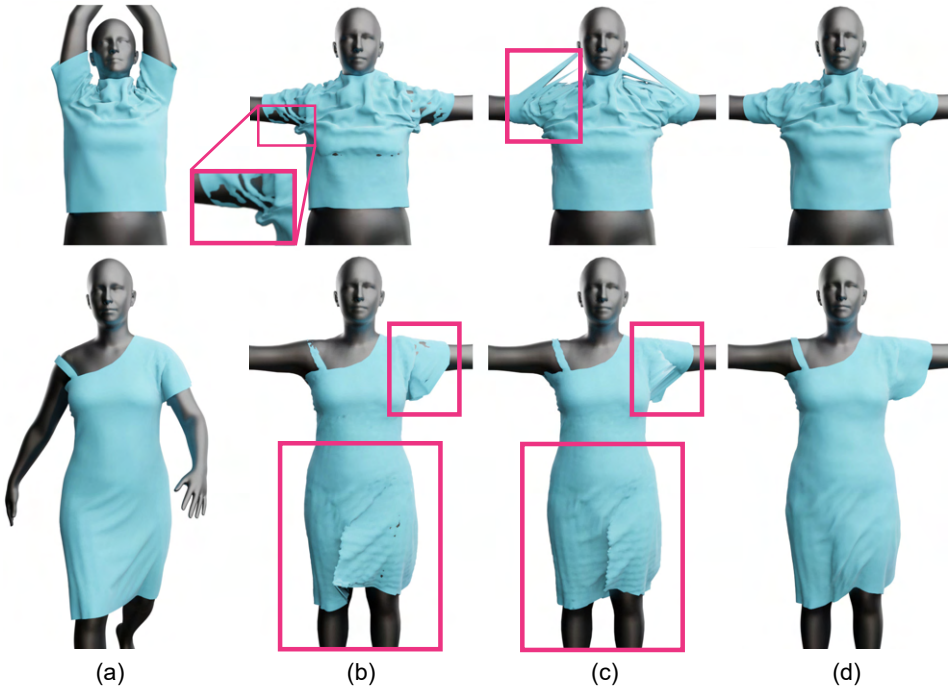


Figure 6.3: Unposing of a T-shirt and a dress in challenging poses: (a) input mesh; (b) unposing with constant weights [PLP20; SOC19], notice the collisions; (c) unposing with variable weights assigned with nearest vertex, it avoids collisions but introduces skinning artifacts and is not temporally stable; (d) unposing with our optimization.

that if a collision-free projection-and-unprojection operation exists, then the learning can be defined entirely in the unposed and deshaped state. This carries many positive properties that we discuss later.

We therefore need a strategy to project ground-truth garments to our canonical space, without introducing collisions. Notice that we cannot use the inverse of Equation 6.7 because the diffused skinning $\widetilde{\mathcal{W}}(\mathbf{X})$ are only defined for unposed shapes. Furthermore, exhaustive search of garment-body nearest vertices for each frame is highly expensive and introduces discontinuities in medial axis areas (see Figure 6.3c).

Therefore, we propose a new optimization-based strategy to find the optimal vertex positions of the garment in the canonical space. Formally, given a ground-truth deformed garment mesh M_G (*i.e.,* generated with physics-based simulation) with known pose θ and shape β, we find its unposed and deshaped representation $\mathbf{X}$ by minimizing

$$\min_{\mathbf{X}} \quad \mathcal{E}_{\text{rec}} + \omega_1 \mathcal{E}_{\text{strain}} + \omega_2 \mathcal{E}_{\text{collision}}. \tag{6.9}$$

In the minimization objective, the data term

$$\mathcal{E}_{\text{rec}} = \left\| \mathbf{M}_{\text{G}} - M_{\text{G}}(\mathbf{X}, \beta, \theta) \right\|_2^2 \tag{6.10}$$

aims at reducing the difference between the simulated garment and the canonical representation projected back to the original space. Notice that $M_{\text{G}}(\mathbf{X}, \beta, \theta)$, defined in Equation 6.7, is well defined for any set of 3D vertices $\mathbf{X}$, and it is fully differentiable thanks to the diffused skinning parameters.

The regularization term

$$\mathcal{E}_{\text{strain}} = \left\| \tfrac{1}{2}(F(\mathbf{T}_{\text{G}}(\mathbf{X}, \beta, \theta))^{\top} F(\mathbf{T}_{\text{G}}(\mathbf{X}, \beta, \theta)) - \mathbf{I}) \right\|_2^2 \tag{6.11}$$

penalizes unrealistic deformations. To measure the amount of deformation of each triangle we use the Green-Lagrange strain tensor, which is rotation and translation invariant. F denotes the deformation gradient of each triangle.

Lastly, we include a term to prevent the optimized vertex positions $\mathbf{X}$ from colliding with the underlying body:

$$\mathcal{E}_{\text{collision}} = \max(\epsilon - SDF(\mathbf{X}), 0) \tag{6.12}$$

This term requires computing the distance to the body surface for all vertices of the deformed garment, which is usually modeled with a Signed Distance Field (SDF). We leverage the fact that bodies in our canonical space are represented with a *constant body mesh*, and therefore the SDF is static and can be precomputed. In practice, and inspired by recent works on neural fields [Par*19; AL20; CZ19; Sit*20; Xie*22], we learn the SDF with a shallow fully-connected network that naturally provides a fully differentiable formulation.

To optimize a sequence, we initialize the optimization with the result of the previous frame. This not only accelerates convergence, but also contributes to stabilize the projection over time. For the first frame, we initialize the optimization with the garment template, which is obtained by simulating the garment with the average body model (*i.e.*, θ and β set to zero).

6.3.4 Generative garment deformation subspace

With the garment model defined in Section 6.3.2, and the strategy to project ground-truth data into our canonical space defined in Section 6.3.3, we could train a supervised data-driven method (*e.g.* a neural network) to learn the garment deformation regressor $R()$

defined in Equation 6.1. However, even though our garment model is designed in such a way that the (un)projection operation between canonical space and posed space does not introduce collisions, residual errors in the optimization of the regressor $R()$ could lead to regressed deformed garments $\mathbf{X}$ with body-garment collisions in the canonical space, which would inevitably propagate to the posed space. In fact, this is a common source of collisions in all data-driven methods [Gun*19; PLP20; SOC19; Wan*18].

Our key contribution to address this challenge is to learn a compact subspace for garment deformations that *reliably solves* garment-body interpretations. To do so, we leverage the fact that in our unposed and deshaped canonical representation of garments, the underlying body shape is *constant*, namely, it is a body shape with $\beta = \mathbf{0}$ and $\theta = \mathbf{0}$. This property enables us to train a variational autoencoder (VAE) to learn a *generative* space of garment deformations with a novel self-supervised collision loss term that is independent of the underlaying body and shape, and therefore naturally generalizes to arbitrary bodies. More specifically, we train the VAE with a loss

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{rec}} + \lambda_1 \mathcal{L}_{\text{laplacian}} + \lambda_2 \mathcal{L}_{\text{collision}} + \lambda_3 \mathcal{L}_{\text{KL}}. \tag{6.13}$$

We define the standard VAE reconstruction term as

$$\mathcal{L}_{\text{rec}} = \left\| \mathbf{X} - D(E(\mathbf{X})) \right\|_1, \tag{6.14}$$

where $E()$ and $D()$ are the encoder and decoder networks, respectively. Since $\mathcal{L}_{\text{rec}}$ does not take into account the neighborhood of the vertex, we add an additional loss term that penalizes error between the mesh laplacians [Tau95; Wan*19]

$$\mathcal{L}_{\text{laplacian}} = \left\| \Delta\mathbf{x} - \Delta D(E(\mathbf{X})) \right\|_1 \tag{6.15}$$

To enforce a subspace free of garment-body collisions, we propose the collision term

$$\mathcal{L}_{\text{collision}} = max(\epsilon - SDF(D(E(\mathbf{X})), 0) + max(\epsilon - SDF(D(\bar{\mathbf{X}}_{\text{rand}})), 0) \tag{6.16}$$

where $\bar{\mathbf{X}}_{\text{rand}} \sim \mathcal{N}(0, 1)$. The first term penalizes collisions in the reconstruction of train data. Our fundamental contribution is the second term, $max(\epsilon - SDF(D(\bar{\mathbf{X}}_{\text{rand}})), 0)$, that samples the latent space and, enabled by the deshaped and unposed canonical representation, checks collisions against a *constant body mesh* with a self-supervised strategy (*i.e.*, we do not need ground-truth garments for this term). This key ingredient allows us to exhaustively sample the latent space and learn a compact garment representation that reliably solves garment-body interpenetrations. As already highlighted, since our garment model is designed to not to introduce body-garment collisions in both the projection and unprojection

operations, garment deformations regressed in the generative subspace do not suffer from collisions even in unseen (*i.e.,* test) sequences.

The self-supervised loss is only useful if the values are sampled from the same distribution as the data. For this purpose, we include an additional term $\mathcal{L}_{\text{KL}}$ to enforce a normal distribution in our latent space.

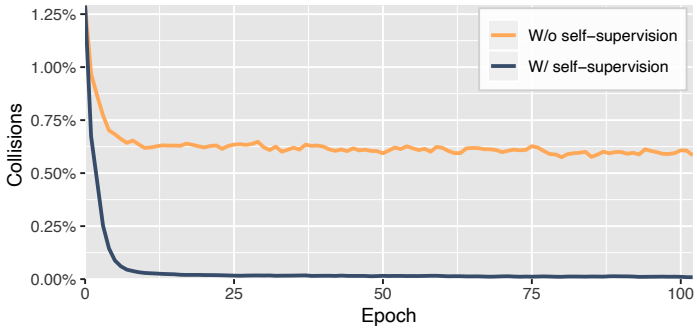


Figure 6.4: Number of body-garment collisions, evaluated in a test set, during the training of the generative subspace. Our novel self-supervised term, described in Equation 6.16, is key to reduce collisions in unseen sequences.

6.4 Regressing garment deformations

Once we have built our generative garment subspace, we encode the ground-truth data and use it to train the recurrent regressor $R(\beta, \gamma)$ from Equation 6.1, which predicts garment deformations as a function of body shape β and motion γ.

Our motion descriptor γ carries information of the current pose as well as its global movement. The off-the-shelf encoding for pose information is to use the joint rotations $\theta \in \mathbb{R}^{72}$ of the underlying human model, but this representation suffers from several problems such as discontinuities, redundant joints, and unnecessary degrees of freedom. Instead, we adopt the compact learned pose descriptor $\bar{\theta} \in \mathbb{R}^{10}$ from Chapter 3, which we found to generalize better. We build the motion vector γ for a given frame by concatenating the descriptor to the velocities and accelerations (computed with finite differences) of the pose, the global rotation $\mathbf{K}$ (represented as Euler angles) and translation $\mathbf{H}$

$$\gamma = \{\bar{\theta}, \frac{\mathrm{d}\bar{\theta}}{\mathrm{d}t}, \frac{\mathrm{d}^2\bar{\theta}}{\mathrm{d}t^2}, \frac{\mathrm{dH}}{\mathrm{d}t}, \frac{\mathrm{d}^2\mathrm{H}}{\mathrm{d}t^2}, \frac{\mathrm{dK}}{\mathrm{d}t}, \frac{\mathrm{d}^2\mathrm{K}}{\mathrm{d}t^2}\} \tag{6.17}$$

The regressor takes as input the motion descriptor $\gamma \in \mathbb{R}^{42}$ and the shape coefficients $\beta \in \mathbb{R}^{10}$ and predicts the encoded garment deformation $\bar{\mathbf{X}}_{\text{pred}} \in \mathbb{R}^{25}$. To learn dynamic effects that depend on previous frames, we use Gated Recurrent Units [Cho*14] as the building blocks of our model. We train using the L1-error of encoded canonical space positions, velocities, and accelerations, which we find improves dynamics compared to optimizing positions alone.

$$\mathcal{L}_{\text{R}} = \mathcal{L}_{\text{pos}} + \rho_1 \mathcal{L}_{\text{vel}} + \rho_2 \mathcal{L}_{\text{acc}} \tag{6.18}$$

6.5 Implementation details

6.5.1 Dataset

We train our model using the same dataset as in Chapter 4, which contains ground-truth simulations of a T-shirt. We also create a new dataset of a dress following the same approach described in Section 4.3.1.

To project the simulation data to the canonical space, we run the optimization described in Section 6.3.3. To this end, we solve Equation (9) with a L-BFGS method and gradients computed with automatic differentiation in TensorFlow [Mar*15]. The weights of the energy function are set to $\omega_1 = 1\text{e}{-4}$, $\omega_2 = 1\text{e}{-2}$ for all meshes.

6.5.2 Neural networks

We use TensorFlow [Mar*15] to implement the networks and Adam [KB15] to train them.

Diffused human model

To generate the training data for the different components of our diffused human model (*e.g.,* blendshape displacements, rigging weights), we exhaustively sample the SMPL model as follows. First, we sample 10,000 points in the bounding box of the template mesh and use libigl [JP*18] to find their closest point in the body surface. Then, from these closest points, we compute ground-truth values (*e.g.*, distances, blendshape displacements, skinning weights, depending on the network we are training) using barycentric interpolation. For

each epoch, we dynamically update the set of training samples. Since the garments can move far from the body surface, we sample points uniformingly to have similar accuracy in all the 3D space. All networks of our diffused human model are trained using a learning rate of 1e-4 and batches of size 256.

- **Signed distance network.** The network that approximates the signed distance field of the body template follows the architecture of SIREN [Sit*20], which uses $sine$ activation functions. The network consists of 5 fully-connected hidden layers with 256 neurons each. We train the network by minimizing the L1 error between the predicted distance and the ground-truth distance.

- **Blendshape networks.** For the networks that approximate the body blendshapes for any arbitrary 3D points, we use a fully-connected network of 5 hidden layers with 256 neurons. We use $ReLU$ activations. We also tried to use SIREN for these networks, but had better results using the architecture of DeepSDF [Par*19].

- **Skinning weight network.** Same as blendshape networks, but the last activation is Softmax (this makes the output always sum 1, which is necessary for skinning weights). Instead of minimizing the L1 norm of the error, we minimize the KL divergence, as done in the work of Liu *et al.* [Liu*19].

Deformation subspace

The generative space of garment deformations is obtained using a Variational Autoencoder (VAE) [KW14]. Both the encoder and the decoder share the same architecture: 4 dense layers of size 2000 with $ReLU$ activations, and layer normalization after each activation. To train the autoencoder, we first train it during 100 epochs with the learning rate set to $1\mathrm{e}{-4}$ and the weights set to $\lambda_1 = 100$, $\lambda_2 = 0$ and $\lambda_3 = 1$ (*i.e.*, we don't optimize collisions during this step). Then, we lower the learning rate to $1\mathrm{e}{-5}$ and we progressively raise the KL and collision losses until we reach the final weights $\lambda_1 = 100$, $\lambda_2 = 10$ and $\lambda_3 = 1$, which are maintained without change until the network converges.

Recurrent regressor

We implement our recurrent regressor $R()$ with 2 GRU layers of size 500. For the recurrent steps we use a $sigmoid$ activation whereas for the final output of each layer we apply the $tanh$ function. We train the network with an initial learning rate of $1\mathrm{e}{-4}$ and batches of 8 sequences, and we set the weights of the velocity and acceleration losses to $\rho_1 = 0.5$ and $\rho_2 = 0.1$, respectively.

6.6 Evaluation

6.6.1 Quantitative evaluation

To quantitatively evaluate the ability of our compact generative subspace to solve body-garment collisions, we show the number of collisions during training in Figure 6.4, evaluated on a *test set* that includes 4 unseen sequences and 17 different shapes. Specifically, we plot an ablation study that shows, in orange, the collisions remaining at each epoch when using only the supervised collision loss (*i.e.*, 1st term of Equation 6.16), and, in black, when also using our self-supervision with the 2nd term of Equation 6.16. The latter dramatically improves the collision handling, and it shows the generalization capabilities of our approach by reaching values close to 0 collisions in unseen sequences.

In Table 6.1 we show a quantitative evaluation of the collisions on test sequences from AMASS [Mah*19] with a total of 53,998 frames and 20 body shapes, and compare our results with the method presented in Chapter 4 [SOC19] and TailorNet [PLP20]. We report the number of collisions for 3 configurations of our method: without the full collision loss, without the self-supervised term, and the full model. All components of our model contribute, leading to a residual of 0.09% with our full model. In contrast, competitive methods suffer from a significantly higher number. These previous methods, without the postprocessing step, generate garment deformations that consistently collide with the underlying body mesh. In contrast, our method directly regresses garment deformations with almost no collisions. Importantly, the primary source of the remaining collisions for our method are self-intersections in the body mesh already present in the AMASS dataset (*e.g.*, a hand interpenetrates the torso).

	TailorNet [PLP20] (w/o postprocess)	Santesteban [SOC19] (w/o postprocess)	Our method (w/o collision loss)	Our method (w/o self-supervision)	Our method
Collisions	5.70%	8.80%	0.62%	0.24%	**0.09%**

Table 6.1: Average number of collisions in 105 test motions from the AMASS dataset [Mah*19].

Finally, in Table 6.2 we quantitatively evaluate our method and compare it to the approach of Chapter 4, which also models dynamics as a function of body shape and motion. To this

end, we use 5 test sequences and 17 shapes, totaling 12,155 frames, and we compute error metrics based on per-vertex Euclidean error, average triangle strain, and average number of collisions. The method presented in this chapter is on par with the per-vertex surface error of the method in Chapter 4, while significantly reducing the number of collisions.

	Chapter 4 [SOC19]	Chapter 6 [San*21]
Error (cm)	**2.9**	3.1
Strain	**0.006**	0.007
Num collisions	80.0	**1.3**

Table 6.2: Quantitative evaluation of our approach in 5 test sequences and 17 body shapes.

6.6.2 Qualitative evaluation

We also qualitatively evaluate the output of our method and compare to recent approaches. Figure 6.6 evaluates the generalization capabilities our method to *unseen body shapes*. Specifically, we interpolate between 2 extremely different real shapes from AMASS [Mah*19], and compare to state-of-the-art data-driven garment models. Importantly, the input shapes are far beyond the range of our training data, therefore here we are also evaluating the *extrapolation* capabilities of the methods. Our method handles such extremely challenging cases very well and does not show visible garment-body collision, while previous methods [SOC19; PLP20] suffer from very noticeable interpenetrations. In Figure 6.5 we show that, although a postprocessing step can effectively mitigate this issue, it can also introduce additional problems.

In Figure 6.7 we evaluate the generalization capabilities of our approach to *unseen motions*, and we compare our results against those of a physics-based simulator. Notice that our method is the first to showcase such a highly-challenging scenario featuring a dress sequence with dynamics.

(a) TailorNet [PLP20] (w/o postprocess) **(b)** TailorNet [PLP20] (w/ postprocess) **(c)** Our method

Figure 6.5: Fixing collisions as a postprocess can introduce undesired bulges, see chest area in (b).

Figure 6.6: Generalization to new shapes. Interpolation between two unseen body shapes (left and right) from the AMASS dataset [Mah*19]. Our deshaped canonical space avoids collisions even in shapes far from the training data.

Figure 6.7: Generalization to new motions. Qualitative comparison with physical simulation [NSO12] (top) in sequence 01_01. Our model (bottom) synthesizes highly realistic dynamics and wrinkles even for challenging unseen motions.

6.6.3 Runtime performance

To evaluate the efficiency of our method, in Table 6.3 we show the runtime performance of each step of our model in a regular desktop PC (AMD Ryzen 7 2700 CPU, Nvidia GTX 1080 Ti GPU, and 32GB of RAM). Our method is capable of generating detailed meshes at high frame rates, even for garments with a high triangle count.

	Triangles	Regressor	Decoder	Projection
T-shirt	8,710	1.7 ms	1.6 ms	1.4 ms
Dress	23,949	1.7 ms	3.5 ms	2.9 ms

Table 6.3: Execution time of each step of our model.

To further validate the advantage of our model with respect to existing methods that apply a postprocessing step to fix the problematic vertices, we compare the runtime performance of both strategies. As we show in Table 6.4, the cost of evaluating the extra networks required by our approach (*i.e.*, the networks of the diffused human model) is significantly lower than the cost of postprocessing required by [PLP20] and [SOC19].

	Triangles	Ours	[SOC19]	[PLP20]
T-shirt	8,710	**1.4 ms**	3.0 ms	210 ms
Dress	23,949	**2.9 ms**	6.9 ms	698 ms

Table 6.4: Evaluation time of the networks required to avoid body-garment collisions (*i.e.*, evaluating the diffused body model to project vertices from canonical to pose space) *vs.* the postprocessing time for [PLP20] and [SOC19] using authors' implementation.

6.7 Conclusions

We have presented the first algorithm to learn garment deformations such that they are essentially collision-free. We have achieved this by designing a novel *unposed* and *deshaped* canonical space built upon two key contributions: a diffused representation of the underlying body and a compact generative subspace of garment deformations. The garment animations produced by our method exhibit a large amount of spatial and temporal detail, and can be inferred extremely quickly, making it suitable for virtual try-on applications.

In the future, we would like to explore using our self-supervised strategy to enforce other physical constraints, such as outputting garments that are free of self-intersections. Moreover, since subspaces are widely used in both data-driven and physics-based methods, we believe that some of our contributions could also be useful beyond garments and virtual try-on.

The main limitation of our method is that, despite providing a robust solution for handling collisions with the body, it does not address collisions between layered garments. Extending this strategy to outfits would require training a model for each garment combination, which would not be feasible in a virtual try-on application with a large number of clothes. To overcome this issue, in the following chapter we present an alternative approach to handle collisions between garments that does not require additional training per outfit, paving the way for accurate, interactive, and scalable virtual try-on systems.

7 Handling collisions between layered garments

The methods presented in Chapters 4, 5 and 6 have shown promising results in modeling garment deformations at interactive framerates. However, these solutions are limited to a single garment layer, and cannot address the combinatorial complexity of mixing different pieces of clothing. Motivated by this limitation, we investigate the use of neural fields for mix-and-match virtual try-on, and identify and solve a fundamental challenge that existing methods do not address: the interaction between layered neural fields. To this end, we propose a model that untangles layered neural fields to represent collision-free garment surfaces. The key ingredient is a neural untangling projection operator that works directly on the layered fields, not on explicit surface representations. Algorithms to resolve object-object interaction are inherently limited by the use of explicit geometric representations, and we show how methods that work directly on neural implicit representations could bring a change of paradigm and open the door to radically different approaches. The method presented in this chapter has led to the following paper, currently under review:

Igor Santesteban, Nils Thuerey, Miguel A. Otaduy, and Dan Casas. "ULNeF: Untangled Layered Neural Fields for Mix-and-Match Virtual Try-On".

7.1 Introduction

The methods presented in previous chapters as well as other state-of-the-art methods [Gun*19; PLP20; Mad*21; BME21; Zha*21a], train a 3D deformation model of one garment (or a predefined outfit) and provide an accurate approximation of physics simulation at runtime, while requiring just a small fraction of the computational cost of physical simulations. However, state-of-the-art virtual try-on is limited to wearing a *single* garment or a predefined outfit, but in real life, we combine many clothes to create different outfits. Unfortunately, existing garment-specific or outfit-specific data-driven solutions cannot

address the combinatorial complexity of mix-and-match virtual try-on. In fact, the problem of mix-and-match virtual try-on poses novel challenges to machine learning algorithms, as it drives the attention toward object *interaction* problems where each object (*e.g.*, each garment) is geometrically complex, and the space of object-object interactions cannot be exhaustively trained.

Object-object interaction has been solved traditionally using explicit geometric representations [NSO12; AE21]. Recent advances in neural models of implicit representations have enabled radically new solutions to many problems, with the ability to efficiently encode parametric models [Sai*21; Den*20; AXS21; Che*21; Pal*21; Hua*20; Wan*21]. However, for object-object interaction, the applicability of neural implicit models is still limited to solving proximity queries against explicit representations, and interaction is actually solved on these explicit representations [Zes*22].

Motivated by the challenges of mix-and-match virtual try-on, we introduce a novel approach to resolve multi-object interaction problems, which works directly on the implicit representations of the objects. We represent multiple possibly colliding surfaces (*e.g.*, multiple garments) using a layered variant of neural fields [Xie*22], and we build upon the untangling operator proposed by Buffet *et al.* [Buf*19] to design an algorithm that untangles these layered neural fields at interactive framerates. In Section 7.2, we present an overview of the existing literature on neural fields. In Section 7.3, we describe how layered neural fields can be parameterized by deformation codes to represent multiple deformable surfaces, and we introduce a neural untangling projection operator that works directly on the layered neural fields, not on geometric surface representations. The result of compositing the neural projection with the layered neural fields is *untangled layered neural fields* (ULNeFs).

In Section 7.4, we show how ULNeFs can be used to efficiently solve mix-and-match virtual try-on for multiple garment layers. As a preprocess, each garment layer is represented using a parametric neural field that is trained for this garment in isolation. At runtime, given a code that represents body shape, we optimize the deformation of each garment using ULNeFs as the key ingredient to resolve collisions. We demonstrate challenging mix-and-match virtual try-on examples with multiple layers of clothing resolved interactively.

In summary, our contribution is threefold:

1. A neural field formulation, based on covariant fields, capable of encoding open surfaces with holes, as well as inside-outside information (Section 7.3.1).

2. A neural projection operator that directly projects entangled surfaces, encoded as neural fields, to untangled configurations, coined as ULNeF (Section 7.3.2).

3. A downstream task using ULNeFs to enable interactive and accurate mix-and-match virtual try-on (Section 7.4).

ULNeFs could see applicability in other object-object interaction problems beyond mix-and-match virtual try-on. Algorithms to resolve object-object interaction are inherently limited by the use of explicit geometric representations. We show how methods that work directly on implicit representations could bring a change of paradigm and enable radically different approaches.

7.2 Neural fields

Over the last few years, neural fields [Xie*22] have emerged as an alternative to the well-established polygonal meshes to represent shapes. Neural fields build on the –also well-established– idea that shapes can be represented as a level set of implicit functions, but propose that such function can be learned with a neural network [CZ19; Mes*19; Mic*21]. These neural representations are compact, continuous, and easily differentiable, which makes them very appealing for a large variety of applications in many fields including Computer Vision [Sai*20; Zhe*21], Computer Graphics [SZW19; Mil*20; Du*21; Yan*21a], and Robotics [Sim*21; Suc*21; Ort*22].

Existing works have leveraged the capabilities of neural fields for a variety of tasks. This has enabled, for example, impressive advances in *reconstructing* clothed humans directly from RGB [Sai*20; He*20; Yan*21b], RGB-D [Li*20b; Su*20; Yu*21; Don*22], and point cloud [CAP20] input. Alternatively, some methods use neural fields to ease the fitting of a parametric human model [Lop*15; JSS18] to sparse inputs [WGT21; Bha*20a; Bha*20b], which yields to detailed 3D reconstructions that can be articulated by the underlying skeleton.

Closer to our work are the methods that use neural fields for *modeling* 3D deformable bodies [Den*20; Nie*19; AXS21; Mih*21; Kar*21; Kar*20] and clothed humans [Che*21; Pal*21; Hua*20; Sai*21; Wan*21]. This is in contrast to previous approaches that tackle these 3D modeling tasks with explicit mesh-based models for humans [San*20; Ma*20; Pon*17], which typically requires accurate surface registrations, and limits the surface details by the mesh topology. A common strategy is to learn dynamic neural fields in a canonical space, reproducing pose-dependent deformations observed in detailed scans [Sai*21; Che*21] or partial depth maps [Pal*21; Wan*21; Don*22]. The learned field is then articulated using forward skinning techniques. Despite the realism of the output deformations, learned fields

encode a *single* surface for clothing and body. In contrast, our formulation defines how to mix and untangle different fields to allow combining multiple garments into a cohesive outfit.

Neural fields have also been used to encode both appearance and volumetric information of a scene, a representation known as Neural Radiance Fields (NeRF) [Mil*20]. Follow up works showed that NeRFs can be used also to encode articulated objects [XAS21; Nog*21; Yan*22] and dynamic scenes [Pum*21; Gaf*21; Par*21]. Specific for humans, A-NeRF [Su*21] transforms NeRF features using a skeleton, and demonstrates that novel motions and viewpoints can be synthesized. NeuralBody [Pen*21] appends learnable features to the vertices of a surface body model, enabling free-viewpoint rendering of animatable humans. Similarly, Kwon *et al.* [Kwo*21] enrich a parametric surface human aggregating spatio-temporal density and color information using transformers. Orthogonal to these works, we do not encode appearance or volume density, but propose a novel formulation to allow the untangled combination of garments encoded using layered fields, which enriches existing representations for humans.

7.3 Untangled layered neural fields

The core of our work is a method that takes as input N neural fields, which implicitly represent N possibly colliding surfaces, and outputs N projected fields –the ULNeFs– which encode collision-free implicit surfaces and minimize the displacement with respect to the input. By using implicit representations, the surfaces are defined as zero-sets of scalar functions. Then, the untangling operation reduces to modifying the scalar functions, which in practice shifts the zero-sets. Figure 7.1 depicts a summary of our main building block.

In this section, we present the untangling operation as an optimization formulated on scalar field values, and we show how this optimization can be efficiently learned with a neural model. Beforehand, we first discuss specifics of the implicit representation of garments, and how we further parameterize the garments as a function of additional settings, in our case body shape.

7.3.1 Implicit surface model

Surfaces can be represented implicitly as the zero-set of their distance field. Formally, given a distance field $f(x), x \in \mathcal{R}^3$, the surface is the set $X = \{x \mid f(x) = 0\}$. However,

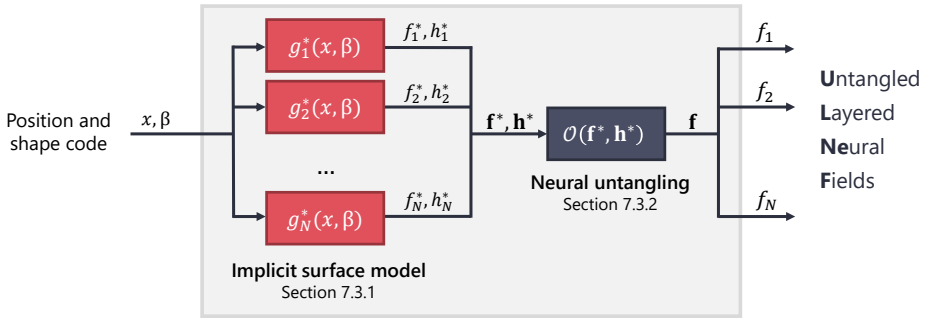


Figure 7.1: Overview of ULNeF.

this implicit representation suffers from two challenging aspects when applied to cloth untangling. First, untangling requires inside-outside information to resolve queries. Second, garments are open surfaces with holes that allow inner layers to pass through, introducing great complexity to the process of collision detection. To the best of our knowledge, we develop the first neural model of garments that addresses these challenges with an implicit representation.

To this end, we represent the garment using two fields: a *signed distance field* $f(x)$ that represents the garment surface and provides a notion of inside-outside and a *covariant field* $h(x)$ that models the volume near the openings that other garments can pass through without producing tangled configurations. We construct the signed distance field by calculating the euclidean distance to the surface and computing the sign as $sign((x-p)\cdot n)$ where p is the closest surface point and n is the normal vector at point p. The covariant field [Buf*19] is computed via a Hermite Radial Basis Function (HRBF) [Wen04] fit by constraining the normals of the seams of cutout regions. Using these fields, we can detect if a point x is in a tangled configuration if $f(x) < 0$ and $h(x) < 0$. Hence, points with $h(x) > 0$ are considered to be unproblematic, as they lie in the volume extending the surface holes.

The implicit surface of a garment g can be represented effectively with this pair of fields $g(x) = (f(x), h(x))$, which we model using a neural network with parameters θ_{fields}. Moreover, using a neural model allows us to further parameterize the surface based on an additional code β, which yields a model $g(x, \beta, \theta_{\text{fields}})$. In our case, we parameterize garment surfaces as a function of body shape [Lop*15], but the implementation could be extended to include other codes such as body pose [SOC19; PLP20; Gun*19] or garment design parameters [Vid*20; SLL20], as in previous learning-based virtual try-on models.

We train the neural implicit model in a supervised way, with loss terms for errors in the fields and their gradients with respect to ground-truth data. Additionally, we encode points

x using Fourier Features [Tan*20], but omit it in the text to simplify the notation. Formally:

$$\theta_{\text{fields}} = \arg\min \quad \mathcal{L}_f + \mathcal{L}_h + \lambda \left(\mathcal{L}_{\frac{\partial f}{\partial x}} + \mathcal{L}_{\frac{\partial h}{\partial x}} \right). \tag{7.1}$$

$$\mathcal{L}_f = \sum_{\beta} \sum_{x} |f(x, \beta, \theta_{\text{fields}}) - f_{\text{GT}}(x, \beta)| \tag{7.2}$$

$$\mathcal{L}_h = \sum_{\beta} \sum_{x} |h(x, \beta, \theta_{\text{fields}}) - h_{\text{GT}}(x, \beta)| \tag{7.3}$$

$$\mathcal{L}_{\frac{\partial f}{\partial x}} = \sum_{\beta} \sum_{x} \left\| \frac{\partial f}{\partial x}(x, \beta, \theta_{\text{fields}}) - \frac{\partial f_{\text{GT}}}{\partial x}(x, \beta) \right\|_1 \tag{7.4}$$

$$\mathcal{L}_{\frac{\partial h}{\partial x}} = \sum_{\beta} \sum_{x} \left\| \frac{\partial h}{\partial x}(x, \beta, \theta_{\text{fields}}) - \frac{\partial h_{\text{GT}}}{\partial x}(x, \beta) \right\|_1 \tag{7.5}$$

In Section 7.5 we provide additional details about the architecture of the neural network, training hyperparameters, and our strategy to sample β and x.

7.3.2 Neural untangling

Let us take as input N possibly colliding implicit surfaces $\{X_i^*\}$ defined by pairs of signed-distance and covariance fields $f_i^*(x), h_i^*(x)$, respectively. Note that the surfaces can be further parameterized by a code β as discussed above. However, we drop this parameterization in this section, as it does not affect the untangling operation. The subindex i denotes the order in which the surfaces should be layered, with surface $i+1$ above, *i.e.*, outside, surface i. We perform untangling by outputting N implicit surfaces $\{X_i\}$ defined by signed distance fields $f_i(x)$. We seek surfaces that are as close as possible to the input surfaces, but remain collision-free.

Thanks to the implicit surface representation, untangling can be formulated as a local operation on the field values at positions $x \in \mathcal{R}^3$. Formally, untangling takes as input two vectors of field values $\mathbf{f}^* = (f_1^*, f_2^*, \dots f_N^*) \in \mathcal{R}^N, \mathbf{h}^* = (h_1^*, h_2^*, \dots h_N^*) \in \mathcal{R}^N$, with components $f_i^* = f_i^*(x), h_i^* = h_i^*(x)$, and it outputs a vector of field values $\mathbf{f} = (f_1, f_2, \dots f_N) \in \mathcal{R}^N$. We denote the local untangling operation as $\mathbf{f} = \mathcal{O}(\mathbf{f}^*, \mathbf{h}^*)$. Applying the local untangling operation at positions x, we obtain the untangled layered field representation $\mathbf{f}(x)$, as shown schematically in Figure 7.1.

The definition of the local untangling operation borrows from the method by Buffet *et al*. [Buf*19]. We define this operation as the following optimization:

$$\mathbf{f} = \mathcal{O}(\mathbf{f}^*, \mathbf{h}^*) = \arg\min \left\| \mathbf{f} - \mathbf{f}^* \right\|_2^2 + \sum_i \sum_{j>i} H(f_i, h_i^*, f_j, h_j^*), \tag{7.6}$$

$$H(f_i, h_i^*, f_j, h_j^*) = \begin{cases} \infty & \text{if } f_i < 0 \text{ and } h_i^* < 0 \text{ and } f_j > 0 \text{ and } h_j^* < 0 \\ 0 & \text{otherwise} \end{cases}$$

In a nutshell, this optimization returns the closest collision-free field values. The collision loss $H()$ penalizes the total loss when a point is outside the top surface j and inside the bottom surface i. Buffet *et al*. [Buf*19] designed an algorithm of complexity $O(N^3)$ to solve the local untangling operation.

Instead, we propose a neural model with parameters θ_{untangl} that *learns* the untangling operation. Then, $\mathcal{O}(\theta_{\text{untangl}})$ can be regarded as a projection operator that projects colliding field values to the closest collision-free values, the ULNeFs $\mathbf{f}$. Importantly, note that this neural projection operator is trained only once for any arbitrary combination of N surfaces, as it operates on the field values, not on the actual surfaces. Hence, once trained, this model naturally generalizes to unseen garments at train time. We train the neural projection operator in a supervised way, based on ground-truth projection examples:

$$\theta_{\text{untangl}} = \arg\min \sum \left\| \mathcal{O}(\mathbf{f}^*, \mathbf{h}^*, \theta_{\text{untangl}}) - \mathbf{f}_{\text{GT}} \right\|_1 \tag{7.7}$$

Please check Section 7.5 for details about training data, architecture and parameters.

7.4 Mix-and-match virtual try-on

Using the ULNeFs presented in the previous section, we now describe how we solve the problem of mix-and-match virtual try-on. The input to the virtual try-on problem consists of neural parametric models of garments trained for each garment in isolation, together with a value of the parametric code to be evaluated (in our case, body shape β). We describe an optimization problem that takes the per-garment models and finds untangled collision-free garments with minimal deformation. The central ingredient of this optimization is the fast evaluation of ULNeFs, as we search for the optimum. Figure 7.2 depicts our pipeline.

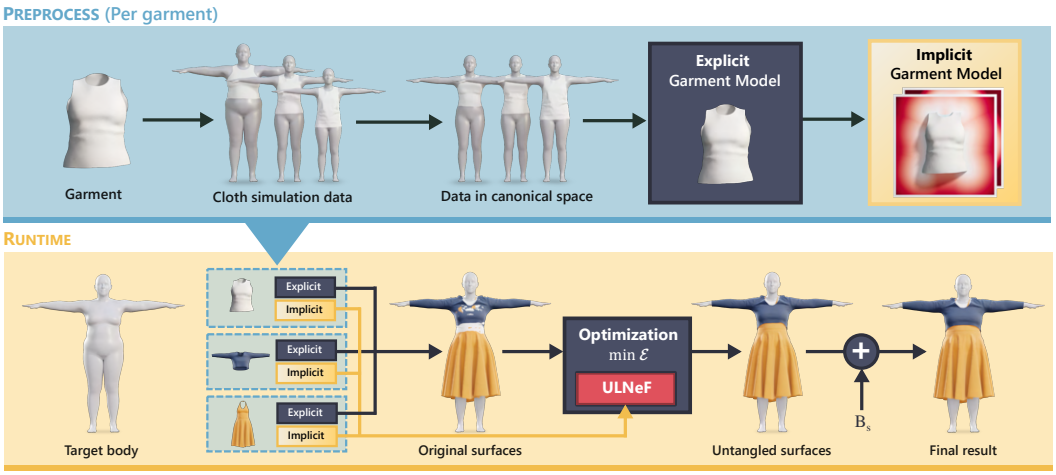


Figure 7.2: Pipeline of our method for mix-and-match virtual try-on. We first preprocess a dataset of garments by simulating each of them in a variety of human shapes. Then, we transform garments into a canonical space, and learn shape-dependent explicit and implicit models. At runtime, we infer explicit and implicit shape-dependent garment deformations, use ULNeF to untangle the implicit representations, and optimize the explicit surfaces to fit into the resulting untangled fields.

To ease the problem, we represent garment deformations in the canonical space from Chapter 6. We start by describing the body model and the canonical-space garment deformation, and then present the optimization of untangled garments.

7.4.1 Explicit garment model

Our explicit garment model builds on top of the parametric SMPL body model [Lop*15] and uses an explicit geometry (*i.e.,* vertices and triangles) to represent garment deformations. SMPL defines a template surface $\mathbf{T}_\mathrm{B}$, local shape- and pose-dependent deformations with respect to this template, and provides a skinning transformation to world space. In this chapter, we have limited our implementation to shape-dependent transformations; therefore, we omit pose and skinning transformation. In SMPL, a point on the body surface M_B for shape β is defined as:

$$M_\mathrm{B}(x, \beta) = x + B_\mathrm{s}(x, \beta), \quad x \in \mathbf{T}_\mathrm{B}, \tag{7.8}$$

where B_s represents a deformation modeled using shape-dependent blend shapes.

To represent garment deformations, we use the canonical garment model introduced in Chapter 6. This canonical model diffuses surface body properties (in this case, the shape-dependent blendshape deformations $\widetilde{B}_\mathrm{s}$) beyond the body surface, to any point in $\mathcal{R}^3$. This

diffusion strategy allows us to retrieve accurate per-point shape-dependent deformations. Hence, following the same formulation as for the body surface, a point on the garment surface M_{G} is obtained by transforming the garment in canonical space $\mathbf{X}$:

$$M_{\text{G}}(x, \beta) = x + \widetilde{B}_{\text{s}}(x, \beta), \quad x \in \mathbf{X}(\beta), \tag{7.9}$$

where $\widetilde{B}_{\text{s}}(x, \beta)$ is the diffused shape blendshape that outputs per-point 3D deformations as a function of the point $x \in \mathcal{R}^3$ and shape parameter β. $\mathbf{X}(\beta)$ is the actual deformed garment, *i.e.,* the vertices of a 3D mesh that exhibit shape-dependent folds and wrinkles, obtained for example with a garment regressor trained on cloth simulation data. Section 7.5 provides more details about the implementation of this regressor.

The garment model is capable of producing accurate and fast deformations of a single garment, but combining the output of multiple models results in deeply tangled surfaces. In the following section, we describe our approach to solve this issue by leveraging ULNeFs.

7.4.2 Optimization of untangled garments

To obtain untangled garment surfaces $\{\mathbf{X}_i(\beta)\}$ for a specific shape code β, we reconstruct the zero-sets of ULNeFs. Note that ULNeFs define implicit surfaces $\{X_i(\beta)\}$, and here we search for explicit mesh-based discretizations. Since ULNeFs are defined by neural fields, during the preprocessing step of each garment we follow the approach presented in Section 7.3.1 to train an implicit equivalent of the explicit garment model.

To reconstruct the zero-sets of ULNeFs, first we initialize possibly colliding garments $\{\mathbf{X}_i^*(\beta)\}$ using the per-garment explicit models. We have observed that just projecting mesh vertices to the zero-sets could yield large triangle distortions. Therefore, when searching for the untangled garment surfaces, we add a penalty term to minimize triangle distortion. Formally, we obtain each untangled garment surface by solving the following optimization:

$$\mathbf{X}_i(\beta) = \arg\min \quad \mathcal{E}_{\text{projection}} + \omega\, \mathcal{E}_{\text{strain}} \tag{7.10}$$

$$\mathcal{E}_{\text{projection}} = \sum_{x \in \mathbf{X}_i(\beta)} f_i(x, \beta)^2, \tag{7.11}$$

$$\mathcal{E}_{\text{strain}} = \sum_{T \in \mathbf{X}_i(\beta)} \left\| \tfrac{1}{2}(F(T)^\top F(T) - \mathbf{I}) \right\|_2^2. \tag{7.12}$$

In the $\mathcal{E}_{\text{projection}}$ term, we evaluate the untangled field f_i for all vertices x in the garment mesh. Note that this requires first evaluating per-garment fields, followed by the neural projection, as shown in Figure 7.1. In the $\mathcal{E}_{\text{strain}}$ term, we evaluate the squared Frobenius norm of Green strain for all triangles T in the garment mesh, with F the deformation gradient.

We solve the optimization using L-BFGS. We have observed that initialization with the per-garment meshes $\{\mathbf{X}_i^*(\beta)\}$ is key for fast convergence. Note also that the gradient computation requires the gradient of the ULNeFs, which is easily obtained thanks to the automatic differentiation capabilities of machine learning frameworks.

7.5 Implementation details

7.5.1 Per-garment preprocess

In total, we train 5 garment models using garments from the Berkeley Garment Library [WOR11; NSO12]. Each garment model consists of an explicit model that predicts the deformed geometry and an implicit model that estimates the signed distance field and covariant field conditioned to body shape. Table 7.1 shows the preprocessing cost per garment.

	Vertices	Triangles	Simulation	Training (explicit)	Training (implicit)
T-shirt	4424	8710	1h 53min	54s	1h 2min
Top	4306	8454	43min	52s	1h 1min
Tank	3010	5825	1h 11min	50s	54min
Pants	3893	7696	50min	52s	56min
Dress	14297	28168	3h 42min	1min 29s	2h 36min

Table 7.1: Preprocessing time per garment.

All our models are implemented in PyTorch, using Adam [KB15] for training, and a linear learning rate scheduler that reduces the initial learning rate by a factor of 0.001 by the end of the training. Next, we define the aspects that are specific to each model.

Explicit garment model

The explicit garment model receives $\beta \in \mathcal{R}^2$ as input and produces the garment deformation $\mathbf{X}(\beta) \in \mathcal{R}^{|V|\times 3}$ in canonical space, were $|V|$ is the number of vertices of the garment mesh. We use the two first coefficients of the SMPL model [Lop*15] since these are enough to capture the largest body deformations. To train the regressor $\mathbf{X}(\beta, \theta_{\text{explicit}})$, where θ_{explicit} are the trainable parameters, we minimize the difference w.r.t ground-truth data:

$$\theta_{\text{explicit}} = \arg\min \sum_{\beta} \left\| \mathbf{X}(\beta, \theta_{\text{explicit}}) - \mathbf{X}_{\text{GT}}(\beta) \right\|_2^2 \tag{7.13}$$

Data generation. We compute ground-truth garment deformations using the cloth simulator ARCSim [NSO12; NPO13]. To this end, first we sample the space of body shapes by selecting 11 evenly-spaced values in $[-2.5, 2.5]$ and then we simulate the garments for each possible combination of those values. With two shape coefficients, this yields 121 body shapes. To simulate, we use the `gray-interlock` material included in the cloth simulator that models an interlock knit fabric made of 60% cotton and 40% polyester. Since the simulations require a collision-free initial state, we start all the simulations using the mean body shape (for which we have a manually created collision-free configuration) and perform 20 interpolation steps towards the target body shapes. Then we simulate for 200 additional steps or until the garment reaches equilibrium.

Data preprocessing. The preprocessing consists on projecting the simulation data to the canonical space proposed in Chapter 6, which removes the influence of body shape in the garment deformation. For example, the height of the avatar introduces a translation in the vertical axis that results in completely different vertex positions, even if the overall deformation remains similar. Removing the influence of body shape is a way of removing this undesired variance in the training data. To make the learning task easier, we also normalize the data per-vertex by subtracting the mean and dividing by the standard deviation.

Network architecture and training. The function $\mathbf{X}(\beta, \theta_{\text{explicit}})$ is modeled as a Multilayer Perceptron (MLP) network with 3 hidden layers of size 256 and ReLU activations. The output layer has size $3|V|$ and after reshaping and denormalizing we obtain the deformed garment $\mathbf{X}(\beta, \theta_{\text{explicit}})$ in canonical space. We train the model for 1000 epochs using batches of size 8 and an initial learning rate of 1e-3.

Implicit garment model

The implicit garment model receives the body shape coefficients $\beta \in \mathcal{R}^2$ and a point $x \in \mathcal{R}^3$ and returns the value of $f(x)$ and $h(x)$ (*i.e.*, the signed distance field and the covariant field evaluated at point x). Our goal is to train an implicit model that is consistent with the explicit model for all body shapes.

Data generation. To generate the ground-truth data for the implicit network, at the beginning of each epoch we randomly sample 20 body shapes from $\mathcal{U}(-2.5, 2.5)$, evaluate the explicit garment regressor to obtain the deformed garment surfaces and, for each surface, we compute ground-truth values of $f(x)$, $h(x)$ and their gradients for all the vertices of the surface as well as 3000 points sampled randomly in the volume (in our implementation, the volume is the bounding-box of the garment and the sampling is done uniformly along each axis). In total, the dataset has $20(|V| + 3000)$ samples. Since the cost of computing the dataset is similar to the cost of a training epoch, while the network trains on the GPU we regenerate the dataset on the CPU. This way we can sample the input space exhaustively and enforce consistency between the implicit and explicit models for any body shape, not just the 121 body shapes used to train the deformation regressor.

Network architecture and training. The implicit garment model is implemented as a MLP network with 4 hidden layers of size 256 and ELU activation functions. The input 3D position x is mapped to a higher dimensional space using Fourier Features [Tan*20]. The mapping is computed as $\gamma(x) = [cos(2\pi \mathbf{B}x), sin(2\pi \mathbf{B}x)]$ where $\mathbf{B}$ is a random Gaussian matrix of size 64×3 whose values are sampled from $\mathcal{N}(0, 2)$. The model is trained for 1000 epochs using batches of size 516 and an initial learning rate of 1e-3. The weight λ of the gradient loss terms is set to 0.1.

7.5.2 Untangling operator.

Data generation. To generate training data for the untangling operator we sample random values of $f^* \in \mathcal{R}^N$ from $\mathcal{U}(-0.2, 1.5)$ and $h^* \in \mathcal{R}^N$ from $\mathcal{U}(-1.0, 1.0)$. For each pair of f^* and h^* we compute ground-truth values of the untangled surfaces f using the method by Buffet *et al.* [Buf*19]. In total, the training set of the untangling operator has 1 million samples.

Network architecture. The untangling operator is implemented as a MLP network with 4 hidden layers of size 256 and ELU activation functions. Since the MLP requires a fixed

input size, we set N=7 so that it can handle up to 7 garment layers, which is more than enough for common outfits. To untangle outfits with less than N layers we simply set $h = 1$ for all the unused slots. The model is trained for 3000 epochs using batches of size 516 and an initial learning rate of 1e-3.

7.5.3 Optimization.

We solve the optimization of the untangled garments using Pytorch's implementation of L-BFGS, with the step size set to 1.0, history size to 100, and line search activated for additional robustness (strong Wolfe method). Empirically, we set $\omega = 1e - 5$ so that the optimization avoids large triangle distortions without interfering with our main goal of moving the vertices to the untangled surfaces. For the results shown Section 7.6, we run the optimization until convergence. For interactive applications, we found that running the optimization for just 4 steps is enough to resolve most collisions and achieve interactive frame rates. We address residual collisions using the rendering solution from [De *10], which applies a small offset to the depth buffer of the outer layer. This solution only works for very small collisions, as large depth offsets result in very noticeable artifacts.

7.6 Evaluation

7.6.1 Quantitative evaluation

In Table 7.2, we present an ablation study of the different terms and encodings used to train the implicit representation for open surfaces described in Section 7.3.1. For each ablation, we show the error of the two fields used in our representation. Results demonstrate that both the encoding of input points with Fourier Features [Tan*20] and the supervision of the gradients contribute to the overall accuracy of the model.

	Ours		W/o Fourier feats.		W/o gradient supervision	
	f	h	f	h	f	h
Error (T-shirt)	**1.1mm**	**0.8mm**	2.0mm	1.0mm	5.3mm	0.9mm
Error (Dress)	**1.3mm**	2.0mm	1.6mm	2.0mm	6.6mm	**1.9mm**

Table 7.2: Ablation study of the different aspects of our implicit surface model.

Table 7.3 evaluates the runtime performance of our approach. Specifically, we compare the evaluation time of the untangling operator of Buffet *et al.* [Buf*19] (*i.e.,* solving Equation 7.6) vs. a forward pass of our learned projection operator. It demonstrates that for complex outfits with thousands of vertices (the outfits shown in Figure 7.4 range from 15k to 30k vertices), our approach runs up to two order of magnitude faster. Similarly, our formulation to evaluate the fields f and h is also significantly faster.

	Untangling operator		Field evaluation	
Nº vertices	Buffet *et al.* [Buf*19]	Ours	Buffet *et al.* [Buf*19]	Ours
1	0.04 ms	0.24 ms	0.08 ms	0.36 ms
5000	81.6 ms	0.68 ms	2.08 ms	0.77 ms
15000	238.5 ms	1.74 ms	6.02 ms	2.00 ms
30000	508.0 ms	3.35 ms	12.1 ms	3.92 ms

Table 7.3: Comparison of runtime performance of the main components of ULNeF. We use the authors' implementation to compare the performance of the untangling operator, and an efficient GPU reimplementation to compare the fields. This comparison was conducted in a regular desktop PC equipped with an AMD Ryzen 7 2700 CPU, an Nvidia GTX 1080 Ti GPU, and 32GB of RAM.

7.6.2 Qualitative evaluation

Figure 7.3 presents a qualitative ablation study of the different terms used to learn our implicit garment models. We show that using Fourier Features [Tan*20] to encode points, as well as supervising the field gradients is required to learn accurate neural fields. In Figure 7.4 we show qualitative results of mix-and-match virtual try-on. For each example, we show the entangled result that state-of-the-art methods [San*21; San*21; PLP20] produce when predicting the deformations of multiple garments, without any postprocess.

(a) Ground-truth **(b)** Ours **(c)** W/o Fourier feats. **(d)** W/o gradient loss

Figure 7.3: Qualitative ablation study of our implicit garment model described in Section 7.3.1. For this particular figure, we use marching cubes to extract the surface.

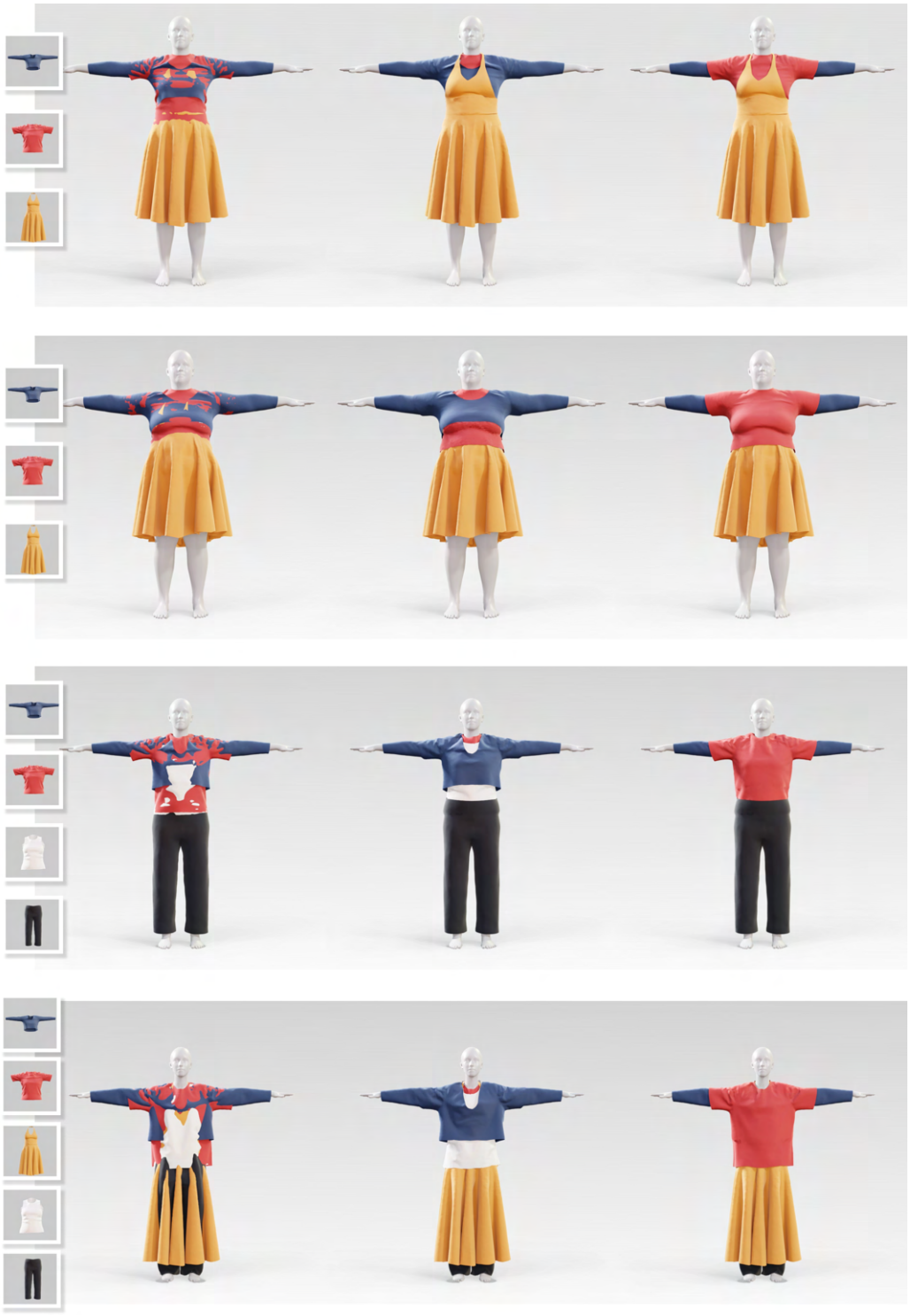


Figure 7.4: Given a set of garments (left insets), existing virtual try-on methods [SOC19] infer their fit into a target body shape but produce a heavily entangled results (left). In contrast, ULNeF untangles the garments by directly projecting their neural fields into a collision-free configuration. Since ULNeF allows to specify the desired order, different outfits can be created (center and right).

7.7 Conclusions

Motivated by the inability of state-of-the-art methods to deal with multiple garments, we have presented Untangled Layered Neural Fields (ULNeF), a novel neural approach to project entangled implicit surfaces to an untangled configuration. The zero-set of the projected fields is collision-free and minimizes the difference with respect to the input surfaces (*i.e.*, maintains the fine-scale details). Importantly, ULNeFs generalize to neural fields unseen at train time and are highly efficient to evaluate.

We have demonstrated the applicability of ULNeFs in mix-and-match virtual try-on, where we leverage the untangled neural fields to resolve collisions between layered garments at interactive frame rates. Our method lets the user combine multiple garments into an outfit and adjust the body shape to obtain accurate estimations of how the outfit will look on them. We address garment untangling via an optimization that preserves the original mesh topology, but we want to stress that if a fixed mesh topology is not a requirement, marching cubes could also be used to recover the untangled meshes directly from the ULNeFs.

Regarding the limitations, our approach for virtual try-on using ULNeFs has only been validated with garments in T-pose. The root of this limitation is the difficulty in extending the formulation based on covariant fields to more complex poses, but we believe that representing garments in the unposed canonical space from Chapter 6 could be helpful to circumvent this issue. Additionally, although we achieve a significant speed-up compared to previous works [Buf*19], our overall runtime is in the order of 200ms per frame. While this is good enough for interactive mix-and-match virtual try-on applications in a static pose, it falls short of producing real-time animations of untangled outfits. Hence, further improvements towards reducing the computational cost remain open avenues for future works.

All in all, we believe ULNeF makes an important step towards modeling interactions of neural fields. Future works could also explore the use of ULNeF in other scenarios that need to account for contact (*e.g.,* hand-object interaction) and are currently limited by explicit surface models.

8 Conclusions

Throughout this thesis we have addressed important challenges in the context of virtual try-on; from estimating soft-tissue dynamics (Chapter 3) and garment deformations (Chapters 4, 5), to handling contact with the body (Chapter 6) and between layered garments (Chapter 7). In this last chapter, we turn our attention to the bigger picture and discuss how we can combine these technologies, what limitations remain open for future works, and where our work fits within the collective effort toward building mainstream virtual try-on applications.

8.1 General discussion

When we started this journey, pioneering works in data-driven garment animation had already shown promising results in modeling garment deformations at highly interactive framerates [Wan*10; De *10; Gua*12; Kim*13; Xu*14; Yan*18; LCT18]. The method presented in Chapter 4 was the first learning-based model to predict accurate garment fit and generalize to multiple body shapes and motions, proving that it is feasible to address virtual try-on from a data-driven perspective. Since then, many great works have pushed the state-of-the-art to new limits [Gun*19; Wan*19; PLP20; Ma*20; Vid*20; Cor*21; Ber*21; BME21].

Learning-based methods are most effective when there is a strong correlation between garment deformation and the underlying body. Unfortunately, this is not the case for loose clothing (*e.g.*, skirts, dresses) where the deformation is driven mostly by cloth dynamics. Consequently, existing methods struggle greatly with loose-fitting garments, and there is a very noticeable degradation in quality as garments become looser (*e.g.*, when trying a big garment on a thin body). This problem has not gone unnoticed by the research community, since recent works are making great improvements in this regard [Zha*21a; Pan*22].

A problem that has not received as much attention is contact modeling. Contact estimation and collision resolution are fundamental for virtual try-on, but traditional collision resolution methods are computationally intensive and not always applicable in data-driven frameworks.

Chapter 6 presents a method that addresses collisions with the body at training time and removes the need for any collision handling at runtime. Chapter 7 presents a method that untangles layered garments at interactive framerates and naturally generalizes to arbitrary garment combinations, a milestone toward mix-and-match virtual try-on. While these methods are effective at solving interpenetrations and presenting collision-free results, they do not capture the real physical response induced by contact. For example, our models assume that the body never deforms under the influence of clothing, when in reality, clothing can greatly affect the shape of the body. Addressing this limitation would enable us to model two-way interactions between deformable surfaces and unify our work on soft-tissue (Chapter 3) and garment (Chapter 4) deformations. It would also improve the fit estimation of certain garments such as corsets or skintight clothing, where soft-tissue undergoes significant deformations as a result of the pressure exerted by the clothes.

Overall, we envision a virtual try-on system in which each garment undergoes an efficient preprocess (Chapter 5) and, at runtime, the user can choose arbitrary combinations of garments and view instantly how those garments will fit (Chapters 4, 6, 7). Thanks to the methods developed in this thesis, we have built an interactive mix-and-match application that serves as a proof of concept of our ideas (Figure 8.1). The current implementation is limited to a static body pose but we hope to keep expanding its scope by incorporating our work on animated avatars and dynamic garments.

Figure 8.1: Screenshot of our interactive mix-and-match demo. Despite being limited in scope, this demo entails significant technical challenges that state-of-the-art methods cannot address. The left view represents the results obtained after doing mix-and-match of state-of-the-art data-driven models, which are trained per garment but cannot be mixed together. The right view shows the results obtained with our method for efficient contact resolution, which handles highly challenging cases at interactive frame rates.

While this thesis makes significant contributions toward building accurate, interactive, and scalable virtual try-on applications, it also rests on some assumptions that narrow the complexity of the real problem. Throughout this thesis we have used physics-based simulation as the ground-truth for our methods and, while some physics-based models have been validated on small pieces of fabric [WOR11; Mig*12; Mig*13], there is little evidence to support that these methods are accurate on a larger scale. In fact, garments contain many elements such as pockets, seams, and buttons that are usually not accounted for during simulation and are added as a postprocess. We hope that the ongoing research on the digitization of garments [Pon*17; Xia*20] and fabrics [Spe*22] will help bridge the gap between real and simulated garments. We have also avoided the complexity of building accurate avatars of real people by working directly with parametric human models, but recent works show promising results in building accurate 3D avatars without expensive scanning setups [Omr*18; Pav*19; Fen*21].

8.2 Final remarks

In a matter of just a few years, we have witnessed huge progress in solving some of the most challenging problems of virtual try-on, and we are optimistic that sooner than later these technologies will be ready for the general public. In addition to developing virtual try-on methods that are accurate and convenient, we have also put great effort into developing methods that account for the diversity of the human body. A diversity that is not always recognized by an industry that gravitates toward unrealistic standards of beauty. We also hope that the development of accurate digital tools for the fashion industry will be pivotal to reduce waste and address the serious environmental impact of this industry.

On a personal note, it is difficult to find words to describe these last years. They have been intense, they have been enriching, and they have also been extremely exhausting. Writing this chapter has been an opportunity to look back and appreciate all the things we have achieved, and all the effort that has gone into it. Time will decide what becomes of this work, but regardless of that, I end this journey feeling grateful for the things I have learned, the people I have met, and all the good moments I have shared with them. Thank you.

Bibliography

[All*19] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. “Learning to Reconstruct People in Clothing from a Single RGB Camera”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2019.

[All*18] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. “Detailed Human Avatars from Monocular Video”. In: *Proc. of International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 98–109.

[AXS21] Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. “imGHUM: Implicit Generative Models of 3D Human Shape and Articulated Pose”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[All*03] Brett Allen, Brian Curless, Brian Curless, and Zoran Popović. “The space of human body shapes: reconstruction and parameterization from range scans”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 22.3 (July 2003), pp. 587–594.

[ACP02] Brett Allen, Brian Curless, and Zoran Popović. “Articulated Body Deformation from Range Scan Data”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 21.3 (2002), pp. 612–619.

[AE21] Sheldon Andrews and Kenny Erleben. “Contact and Friction Simulation for Computer Graphics”. In: *ACM SIGGRAPH 2021 Courses*. SIGGRAPH ’21. 2021.

[Ang*05] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, et al. “SCAPE: Shape Completion and Animation for PEople”. In: *Proc. of ACM SIGGRAPH*. 2005, pp. 408–416.

[AL20] Matan Atzmon and Yaron Lipman. “SAL: Sign Agnostic Learning of Shapes from Raw Data”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2020.

[Bar*16] Aric Bartle, Alla Sheffer, Vladimir G. Kim, et al. “Physics-Driven Pattern Adjustment for Direct 3D Garment Editing”. In: *ACM Transactions on Graphics* 35.4 (July 2016).

[BMM17] Jan Bender, Matthias Müller, and Miles Macklin. “A survey on position based dynamics”. In: *Eurographics Tutorials*. 2017.

[Ben*14] Jan Bender, Matthias Müller, Miguel A Otaduy, Matthias Teschner, and Miles Macklin. “A Survey on Position–Based Simulation Methods in Computer Graphics”. In: *Computer Graphics Forum* 33.6 (2014), pp. 228–251.

[BME20] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. “CLOTH3D: Clothed 3D Humans”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2020.

[BME21] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. “PBNS: Physically Based Neural Simulation for Unsupervised Garment Pose Space Deformation”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 40.6 (Dec. 2021).

[Ber*21] Hugo Bertiche, Meysam Madadi, Emilio Tylson, and Sergio Escalera. “DeePSD: Automatic Deep Skinning and Pose Space Deformation for 3D Garment Animation”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[Bha*03] Kiran S. Bhat, Christopher D. Twigg, Jessica K. Hodgins, et al. “Estimating Cloth Simulation Parameters from Video”. In: *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2003, pp. 37–51.

[Bha*20a] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. “Combining Implicit Function Learning and Parametric Models for 3D Human Reconstruction”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2020.

[Bha*20b] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. “LoopReg: Self-supervised Learning of Implicit Surface Correspondences, Pose and Shape for 3D Human Mesh Registration”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[Bha*19] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. “Multi-garment net: Learning to Dress 3D People from Images”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2019, pp. 5420–5430.

[Bog*16] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, et al. “Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image”. In: *Computer Vision – ECCV 2016*. Lecture Notes in Computer Science. Springer International Publishing, Oct. 2016.

[Bog*17] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. “Dynamic FAUST: Registering Human Bodies in Motion”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2017.

[Bou*14] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. “ Projective Dynamics: Fusing Constraint Projections for Fast Simulation”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 33.4 (2014), pp. 1–11.

[Bou*13] Katherine L Bouman, Bei Xiao, Peter Battaglia, and William T Freeman. “Estimating the Material Properties of Fabric from Video”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 1984–1991.

[Bra*08] Derek Bradley, Tiberiu Popa, Alla Sheffer, Wolfgang Heidrich, and Tamy Boubekeur. “Markerless Garment Capture”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 27.3 (2008), p. 99.

[Bro*12] Remi Brouet, Alla Sheffer, Laurence Boissieux, and Marie-Paule Cani. “Design Preserving Garment Transfer”. In: *ACM Transactions on Graphics* 31.4 (2012), 36:1–36:11.

[Buf*19] Thomas Buffet, Damien Rohmer, Loic Barthe, Laurence Boissieux, and Marie-Paule Cani. “Implicit Untangling: A Robust Solution for Modeling Layered Clothing”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*. 2019.

[BNT21] Andrei Burov, Matthias Nießner, and Justus Thies. “Dynamic Surface Function Networks for Clothed Human Bodies”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[CBI10] Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. “Probabilistic deformable surface tracking from multiple videos”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2010, pp. 326–339.

[Cap*02] Steve Capell, Seth Green, Brian Curless, Tom Duchamp, and Zoran Popović. “Interactive Skeleton-driven Dynamic Deformations”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 21.3 (2002), pp. 586–593.

[CO18] Dan Casas and Miguel A. Otaduy. “Learning Nonlinear Soft-Tissue Dynamics for Interactive Avatars”. In: *Proc. of the ACM on Computer Graphics and Interactive Techniques* 1.1 (May 2018).

[Che*21] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. “SNARF: Differentiable Forward Skinning for Animating Non-Rigid Neural Implicit Shapes”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[CLZ13] Yinpeng Chen, Zicheng Liu, and Zhengyou Zhang. “Tensor-based human body modeling”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 105–112.

[CZ19] Zhiqin Chen and Hao Zhang. “Learning Implicit Fields for Generative Shape Modeling”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2019.

[CAP20] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. “Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2020.

[Cho*14] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014.

[Cir*14] Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A Otaduy. “Yarn-Level Simulation of Woven Cloth”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 33.6 (2014), pp. 1–11.

[CLO15] Gabriel Cirio, Jorge Lopez-Moreno, and Miguel A Otaduy. “Efficient simulation of knitted cloth using persistent contacts”. In: *Proc. of ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA)*. 2015, pp. 55–61.

[CMU] CMU. *CMU Graphics Lab Motion Capture Database*.

[Cor*21] Enric Corona, Albert Pumarola, Guillem Alenyà, Gerard Pons-Moll, and Francesc Moreno-Noguer. “SMPLicit: Topology-aware Generative Model for Clothed People”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2021.

[CML21] Aiyu Cui, Daniel McKee, and Svetlana Lazebnik. “Dressing in Order: Recurrent Person Image Generation for Pose Transfer, Virtual Try-On and Outfit Editing”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 14638–14647.

[De *10] Edilson De Aguiar, Leonid Sigal, Adrien Treuille, and Jessica K Hodgins. “Stable Spaces for Real-time Clothing”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 29.4 (2010).

[Den*20] Boyang Deng, John P Lewis, Timothy Jeruzalski, et al. “NASA: Neural Articulated Ahape Approximation”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2020.

[DB13] Crispin Deul and Jan Bender. “Physically-Based Character Skinning”. In: *Workshop on Virtual Reality Interaction and Physical Simulation*. Ed. by Jan Bender, Jeremie Dequidt, Christian Duriez, and Gabriel Zachmann. The Eurographics Association, 2013.

[Don*19] Haoye Dong, Xiaodan Liang, Xiaohui Shen, et al. “Towards Multi-Pose Guided Virtual Try-On Network”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2019, pp. 9025–9034.

[Don*22] Zijian Dong, Chen Guo, Jie Song, et al. “PINA: Learning a Personalized Implicit Neural Avatar from a Single RGB-D Video Sequence”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2022.

[Dou*15] Mingsong Dou, Jonathan Taylor, Henry Fuchs, Andrew Fitzgibbon, and Shahram Izadi. “3D scanning deformable objects with a single RGBD sensor”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 493–501.

[Du*21] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B. Tenenbaum, and Jiajun Wu. “Neural Radiance Flow for 4D View Synthesis and Video Processing”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[FCS15] Andrew Feng, Dan Casas, and Ari Shapiro. “Avatar Reshaping and Automatic Rigging Using a Deformable Model”. In: *Proc. of ACM SIGGRAPH Conference on Motion in Games (MIG)*. 2015, pp. 57–64.

[Fen*21] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael Black. “Collaborative Regression of Expressive Bodies using Moderation”. In: *Proc. of International Conference on 3D Vision (3DV)*. Dec. 2021, pp. 792–804.

[FTP16] Marco Fratarcangeli, Valentina Tibaldo, and Fabio Pellacini. “Vivace: a Practical Gauss-Seidel Method for Stable Soft Body Dynamics”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 35.6 (2016), pp. 1–9.

[Ful*19] Lawson Fulton, Vismay Modi, David Duvenaud, David IW Levin, and Alec Jacobson. “Latent-space Dynamics for Reduced Deformable Simulation”. In: *Computer Graphics Forum (Proc. Eurographics)* 38.2 (2019), pp. 379–391.

[Gaf*21] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. “Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2021.

[Gas*15] Theodore F Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M Teran. “Optimization Integrator for Large Time Steps”. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 21.10 (2015), pp. 1103–1115.

[Ge*21a] Chongjian Ge, Yibing Song, Yuying Ge, et al. “Disentangled Cycle Consistency for Highly-realistic Virtual Try-On”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)* (2021).

[Ge*21b] Yuying Ge, Yibing Song, Ruimao Zhang, et al. “Parser-Free Virtual Try-on via Distilling Appearance Flows”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)* (2021).

[Gua*12] Peng Guan, Loretta Reiss, David A Hirshberg, Alexander Weiss, and Michael J Black. “DRAPE: DRessing Any PErson”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 31.4 (2012).

[Gun*19] Erhan Gundogdu, Victor Constantin, Amrollah Seifoddini, et al. “GarNet: A two-stream network for fast and accurate 3D cloth draping”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2019.

[Hab*21] Marc Habermann, Lingjie Liu, Weipeng Xu, et al. “Real-time Deep Dynamic Characters”. In: *ACM Transactions on Graphics* 40.4 (Aug. 2021).

[Hah*14] Fabian Hahn, Bernhard Thomaszewski, Stelian Coros, et al. “Subspace Clothing Simulation Using Adaptive Bases”. In: *ACM Transactions on Graphics* 33.4 (July 2014), 105:1–105:9.

[Han*19] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R. Scott. “ClothFlow: A Flow-Based Model for Clothed Person Generation”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. Oct. 2019.

[Han*18] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. “Viton: An image-based virtual try-on network”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7543–7552.

[Has*09] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. “A Statistical Model of Human Pose and Body Shape”. In: *Computer Graphics Forum* 28.2 (2009), pp. 337–346.

[HSR13] Stefan Hauswiesner, Matthias Straka, and Gerhard Reitmayr. “Virtual try-on through image-based rendering”. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 19.9 (2013), pp. 1552–1565.

[He*16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[He*16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Identity mappings in deep residual networks”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2016, pp. 630–645.

[He*20] Tong He, John Collomosse, Hailin Jin, and Stefano Soatto. “Geo-pifu: Geometry and pixel aligned implicit functions for single-view human reconstruction”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020, pp. 9276–9287.

[HFE13] Anna Hilsmann, Philipp Fechteler, and Peter Eisert. “Pose space image based rendering”. In: *Computer Graphics Forum* 32.2 (2013), pp. 265–274.

[HSC02] Adrian Hilton, Jonathan Starck, and Gordon Collins. “From 3D shape capture to animated models”. In: *IEEE Conference on 3D Data Processing, Visualisation and Transmission*. 2002, pp. 246–255.

[Hir*12] David A. Hirshberg, Matthew Loper, Eric Rachlin, and Michael J. Black. “Coregistration: Simultaneous Alignment and Modeling of Articulated 3D Shape”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2012, pp. 242–255.

[Hua*20] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. “ARCH: Animatable Reconstruction of Clothed Humans”. In: *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3093–3102.

[IMC20] Thibaut Issenhuth, Jérémie Mary, and Clément Calauzènes. “Do Not Mask What You Do Not Need to Mask: a Parser Free Virtual Try-On”. In: *Proc. of European Conference on Computer Vision (ECCV)* (2020).

[JP*18] Alec Jacobson, Daniele Panozzo, et al. *libigl: A simple C++ geometry processing library*. https://libigl.github.io/. 2018.

[Jai*10] Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. “MovieReshape: Tracking and Reshaping of Humans in Videos”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 29.5 (2010).

[Jin*20] Ning Jin, Yilin Zhu, Zhenglin Geng, and Ron Fedkiw. “A Pixel-Based Framework for Data-Driven Clothing”. In: *Computer Graphics Forum (Proc. of SCA)* (2020).

[JSS18] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. “Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2018.

[Kad*16] Petr Kadleček, Alexandru-Eugen Ichim, Tiantian Liu, Jaroslav Křivánek, and Ladislav Kavan. “Reconstructing Personalized Anatomical Models for Physics-based Body Animation”. In: *ACM Transactions on Graphics* 35.6 (2016), 213:1–213:13.

[KJM10] Jonathan M Kaldor, Doug L James, and Steve Marschner. “Efficient yarn-based cloth with adaptive contact linearization”. In: *Proc of ACM SIGGRAPH*. 2010.

[KJM08] Jonathan M. Kaldor, Doug L. James, and Steve Marschner. “Simulating Knitted Cloth at the Yarn Level”. In: *ACM Transactions on Graphics* 27.3 (2008), pp. 1–9.

[Kan*18] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. “End-to-end Recovery of Human Shape and Pose”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2018.

[Kar*21] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. “A Skeleton-Driven Neural Occupancy Representation for Articulated Hands”. In: *International Conference on 3D Vision (3DV)*. 2021.

[Kar*20] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, et al. “Grasping field: Learning implicit representations for human grasps”. In: *International Conference on 3D Vision (3DV)*. 2020, pp. 333–344.

[Kav*11] Ladislav Kavan, Dan Gerszewski, Adam W. Bargteil, and Peter-Pike Sloan. “Physics-Inspired Upsampling for Cloth Simulation in Games”. In: *Proc. of ACM SIGGRAPH*. 2011.

[Kim*13] Doyub Kim, Woojong Koh, Rahul Narain, et al. “Near-exhaustive precomputation of secondary cloth effects”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 32.4 (2013), pp. 1–8.

[Kim*17] Meekyoung Kim, Gerard Pons-Moll, Sergi Pujades, et al. “Data-Driven Physics for Human Soft Tissue Animation”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 36.4 (2017).

[KB15] Diederick P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *International Conference on Learning Representations (ICLR)*. 2015.

[KW14] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *International Conference on Learning Representations (ICLR)*. 2014.

[KB18] Martin Komaritzan and Mario Botsch. “Projective Skinning”. In: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1.1 (2018), 12:1–12:19.

[Kwo*21] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. “Neural human performer: Learning generalizable radiance fields for human performance rendering”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021.

[LCT18] Zorah Lähner, Daniel Cremers, and Tony Tung. “DeepWrinkles: Accurate and Realistic Clothing Modeling”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2018.

[LCA05] Caroline Larboulette, Marie-Paule Cani, and Bruno Arnaldi. “Dynamic skinning: adding real-time dynamic effects to an existing character animation”. In: *Proc. of Spring Conference on Computer graphics*. ACM. 2005, pp. 87–93.

[Lee*19] Hyug Jae Lee, Rokkyu Lee, Minseok Kang, Myounghoon Cho, and Gunhan Park. “LA-VITON: A Network for Looking-Attractive Virtual Try-On”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV) Workshops*. Oct. 2019.

[Lee*10] Yongjoon Lee, Sung-eui Yoon, Seungwoo Oh, Duksu Kim, and Sunghee Choi. “Multi-Resolution Cloth Simulation”. In: 29.7 (2010), pp. 2225–2232.

[LCF00] John P Lewis, Matt Cordner, and Nickson Fong. “Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation”. In: *Proc. of the Annual Conference on Computer Graphics and Interactive Techniques*. 2000, pp. 165–172.

[Li*17a] Changjian Li, Hao Pan, Yang Liu, Alla Sheffer, and Wenping Wang. “BendSketch: Modeling Freeform Surfaces Through 2D Sketching”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 36.4 (2017), 125:1–125:14.

[Li*18] Jie Li, Gilles Daviet, Rahul Narain, et al. “An Implicit Frictional Contact Solver for Adaptive Cloth Simulation”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 37.4 (2018), pp. 1–15.

[LLK19] Jing Li, Tiantian Liu, and Ladislav Kavan. “Fast Simulation of Deformable Characters with Articulated Skeletons in Projective Dynamics”. In: *Proc. of ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA)*. 2019, 1:1–1:10.

[Li*21] Kedan Li, Min Jin Chong, Jingen Liu, and David Forsyth. “Toward Accurate and Realistic Outfits Visualization with Attention to Details”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)* (2021).

[Li*17b] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. “Learning a Model of Facial Shape and Expression from 4D Scans”. In: *ACM Transactions on Graphics* 36.6 (Nov. 2017).

[Li*20a] Xueting Li, Sifei Liu, Kihwan Kim, et al. “Self-supervised Single-view 3D Reconstruction via Semantic Consistency”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2020.

[Li*20b] Zhe Li, Tao Yu, Chuanyu Pan, Zerong Zheng, and Yebin Liu. “Robust Dd self-portraits in seconds”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 1344–1353.

[Liu*13] Libin Liu, KangKang Yin, Bin Wang, and Baining Guo. “Simulation and Control of Skeleton-driven Soft Body Characters”. In: *ACM Transactions on Graphics* 32.6 (2013), 215:1–215:8.

[Liu*19] Lijuan Liu, Youyi Zheng, Di Tang, et al. “NeuroSkinning: Automatic Skin Binding for Production Characters with Deep Graph Networks”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 38.4 (July 2019).

[LBK17] Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. “Quasi-Newton Methods for Real-Time Simulation of Hyperelastic Materials”. In: *ACM Transactions on Graphics* 36.4 (2017).

[LMB14] Matthew Loper, Naureen Mahmood, and Michael J. Black. “MoSh: Motion and Shape Capture from Sparse Markers”. In: *ACM Transactions on Graphics* 33.6 (2014), 220:1–220:13.

[Lop*15] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. “SMPL: A skinned multi-person linear model”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 34.6 (2015), pp. 1–16.

[Ly*20] Mickaël Ly, Jean Jouve, Laurence Boissieux, and Florence Bertails-Descoubes. “Projective Dynamics with Dry Frictional Contact”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 39.4 (2020).

[Ma*20] Qianli Ma, Jinlong Yang, Anurag Ranjan, et al. “Learning to Dress 3D People in Generative Clothing”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2020.

[Ma*21] Qianli Ma, Jinlong Yang, Siyu Tang, and Michael J Black. “The power of points for modeling humans in clothing”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[Mad*21] Meysam Madadi, Hugo Bertiche, Wafa Bouzouita, Isabelle Guyon, and Sergio Escalera. "Learning Cloth Dynamics: 3D+Texture Garment Reconstruction Benchmark". In: *Proc. of the NeurIPS 2020 Competition and Demonstration Track*. Ed. by Hugo Jair Escalante and Katja Hofmann. Vol. 133. Proceedings of Machine Learning Research. PMLR, Dec. 2021, pp. 57–76.

[Mah*19] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. "AMASS: Archive of Motion Capture as Surface Shapes". In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. Oct. 2019, pp. 5442–5451.

[Mar*11] Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. "Example-Based Elastic Materials". In: *ACM Transactions on Graphics* 30.4 (2011).

[Mar*15] Martín Abadi, Ashish Agarwal, Paul Barham, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

[MBR17] Julieta Martinez, Michael J Black, and Javier Romero. "On human motion prediction using recurrent neural networks". In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017, pp. 4674–4683.

[McA*11] Aleka McAdams, Yongning Zhu, Andrew Selle, et al. "Efficient Elasticity for Character Skinning with Contact and Collisions". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 30.4 (July 2011), 37:1–37:12.

[Mes*19] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. "Occupancy Networks: Learning 3D Reconstruction in Function Space". In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2019.

[Mic*21] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. "Deep level sets: Implicit surface representations for 3d shape inference". In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[Mig*12] Eder Miguel, Derek Bradley, Bernhard Thomaszewski, et al. "Data-Driven Estimation of Cloth Simulation Models". In: *Computer Graphics Forum (Proc. Eurographics)* 31 (2012), pp. 519–528.

[Mig*13] Eder Miguel, Rasmus Tamstorf, Derek Bradley, et al. "Modeling and Estimation of Internal Friction in Cloth". In: *ACM Transactions on Graphics* 32.6 (2013), pp. 1–10.

[Mih*21] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. "LEAP: Learning Articulated Occupancy of People". In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2021.

[Mil*20] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, et al. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *Proc. of European Conference on Computer Vision (ECCV)*. 2020.

[Mon*12] Domen Mongus, Blaz Repnik, Marjan Mernik, and B Žalik. "A hybrid evolutionary algorithm for tuning a cloth-simulation model". In: *Applied Soft Computing* 12.1 (2012), pp. 266–273.

[NPO13] Rahul Narain, Tobias Pfaff, and James F. O'Brien. "Folding and Crumpling Adaptive Sheets". In: *ACM Transactions on Graphics* 32.4 (July 2013), 51:1–51:8.

[NSO12] Rahul Narain, Armin Samii, and James F O'brien. "Adaptive Anisotropic Remeshing for Cloth Simulation". In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 31.6 (2012), pp. 1–10.

[Nea*06] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. "Physically Based Deformable Models in Computer Graphics". In: *Computer Graphics Forum* 25.4 (2006), pp. 809–836.

[NH14] Alexandros Neophytou and Adrian Hilton. "A layered model of human body and garment deformation". In: *Proc. of International Conference on 3D Vision (3DV)*. 2014, pp. 171–178.

[Neu*20] Assaf Neuberger, Eran Borenstein, Bar Hilleli, Eduard Oks, and Sharon Alpert. "Image Based Virtual Try-On Network From Unpaired Data". In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[Neu*13a] Thomas Neumann, Kiran Varanasi, Nils Hasler, et al. "Capture and Statistical Modeling of Arm-Muscle Deformations". In: *Computer Graphics Forum* 32.2 (2013), pp. 285–294.

[Neu*13b] Thomas Neumann, Kiran Varanasi, Stephan Wenger, et al. "Sparse Localized Deformation Components". In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 32.6 (Nov. 2013).

[Nie*19] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. "Occupancy Flow: 4D Reconstruction by Learning Particle Dynamics". In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2019.

[Nog*21] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. "Neural Articulated Radiance Field". In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[NVI18] NVIDIA Corporation. *TensorRT: Programmable Inference Accelerator*. 2018.

[Omr*18] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. "Neural Body Fitting: Unifying Deep Learning and Model-Based Human Pose and Shape Estimation". In: 2018.

[Ort*22] Joseph Ortiz, Alexander Clegg, Jing Dong, et al. "iSDF: Real-Time Neural Signed Distance Fields for Robot Perception". In: *Robotics: Science and Systems (RSS)*. 2022.

[Pai*18] Dinesh K. Pai, Austin Rothwell, Pearson Wyder-Hodge, et al. "The Human Touch: Measuring Contact with Real Human Soft Tissues". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 37.4 (2018), 58:1–58:12.

[Pal*21] Pablo Palafox, Aljaz Bozic, Justus Thies, Matthias Nießner, and Angela Dai. "Neural Parametric Models for 3D Deformable Shapes". In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[Pan*22] Xiaoyu Pan, Jiaming Mai, Xinwei Jiang, et al. "Predicting Loose-Fitting Garment Deformations Using Bone-Driven Motion Networks". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* (2022).

[Par*19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. “DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2019.

[Par*21] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, et al. “Nerfies: Deformable Neural Radiance Fields”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)* (2021).

[PH06] Sang Il Park and Jessica K Hodgins. “Capturing and animating skin deformation in human motion”. In: *ACM Transactions on Graphics* 25.3 (2006), pp. 881–889.

[PH08] Sang Il Park and Jessica K. Hodgins. “Data-driven Modeling of Skin and Muscle Deformation”. In: *ACM Transactions on Graphics* 27.3 (2008), 96:1–96:6.

[PMB13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. “On the difficulty of training recurrent neural networks”. In: *International Conference on Machine Learning (ICML)*. 2013, pp. 1310–1318.

[PLP20] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. “The Virtual Tailor: Predicting Clothing in 3D as a Function of Human Pose, Shape and Garment Style”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2020.

[Pav*19] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, et al. “Expressive Body Capture: 3D Hands, Face, and Body from a Single Image”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10975–10985.

[Pen*21] Sida Peng, Yuanqing Zhang, Yinghao Xu, et al. “Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 9054–9063.

[Pis*17] Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. “Building statistical shape spaces for 3D human modeling”. In: *Pattern Recognition* 67 (2017), pp. 276–286.

[Pon*17] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. “ClothCap: Seamless 4D clothing capture and retargeting”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 36.4 (2017).

[Pon*15] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. “Dyna: A Model of Dynamic Human Shape in Motion”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 34.4 (2015).

[Pum*21] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. “D-NeRF: Neural Radiance Fields for Dynamic Scenes”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2021.

[Raj*18] Amit Raj, Patsorn Sangkloy, Huiwen Chang, et al. “SwapNet: Image Based Garment Transfer”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2018, pp. 666–682.

[Ras*20] Abdullah-Haroon Rasheed, Victor Romero, Florence Bertails-Descoubes, et al. “Learning to Measure the Static Friction Coefficient in Cloth Contact”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2020.

[Rob*17] Nadia Robertini, Dan Casas, Edilson De Aguiar, and Christian Theobalt. “Multi-view Performance Capture of Surface Details”. In: *International Journal of Computer Vision* 124.1 (2017), pp. 96–113.

[RMS19] M Romeo, C Monteagudo, and D Sánchez-Quirós. “Muscle and Fascia Simulation with Extended Position Based Dynamics”. In: *Computer Graphics Forum* (2019).

[Rom*20] Cristian Romero, Miguel A. Otaduy, Dan Casas, and Jesus Perez. “Modeling and Estimation of Nonlinear Skin Mechanics for Animated Avatars”. In: *Computer Graphics Forum (Proc. Eurographics)* 39.2 (2020).

[RTB17] Javier Romero, Dimitrios Tzionas, and Michael J. Black. “Embodied Hands: Modeling and Capturing Hands and Bodies Together”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*. 245:1–245:17 36.6 (Nov. 2017).

[Run*20] Tom F. H. Runia, Kirill Gavrilyuk, Cees G. M. Snoek, and Arnold W. M. Smeulders. “Cloth in the Wind: A Case Study of Physical Measurement Through Simulation”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2020.

[Sai*19] Shunsuke Saito, Zeng Huang, Ryota Natsume, et al. “PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2019.

[Sai*20] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. “PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[Sai*21] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J. Black. “SCANimate: Weakly Supervised Learning of Skinned Clothed Avatar Networks”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2021.

[San*20] Igor Santesteban, Elena Garces, Miguel A. Otaduy, and Dan Casas. “SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Humans”. In: *Computer Graphics Forum (Proc. Eurographics)* 39.2 (2020).

[SOC19] Igor Santesteban, Miguel A. Otaduy, and Dan Casas. “Learning-Based Animation of Clothing for Virtual Try-On”. In: *Computer Graphics Forum (Proc. Eurographics)* 38.2 (2019).

[San*21] Igor Santesteban, Nils Thuerey, Miguel A Otaduy, and Dan Casas. “Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)* (2021).

[SM06] Volker Scholz and Marcus Magnor. “Texture Replacement of Garments in Monocular Video Sequences”. In: *Eurographics Conference on Rendering Techniques*. 2006, pp. 305–312.

[Sch*05] Volker Scholz, Timo Stich, Michael Keckeisen, Markus Wacker, and Marcus Magnor. “Garment Motion Capture Using Color-Coded Patterns”. In: *Computer Graphics Forum* 24.3 (2005), pp. 439–447.

[Sel*09] Andrew Selle, Jonathan Su, Geoffrey Irving, and Ronald Fedkiw. “Robust High-Resolution Cloth Using Parallelism, History-Based Collisions, and Accurate Friction”. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 15.2 (Mar. 2009), pp. 339–350.

[SLL20] Yu Shen, Junbang Liang, and Ming C. Lin. “GAN-based Garment Generation Using Sewing Pattern Images”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2020.

[SB12] Eftychios Sifakis and Jernej Barbic. “FEM simulation of 3D deformable solids: a practitioner’s guide to theory, discretization and model reduction”. In: *SIGGRAPH 2012 Courses*. ACM, 2012, pp. 1–50.

[Sim*21] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, et al. *Neural Descriptor Fields: SE(3)-Equivariant Object Representations for Manipulation*. 2021.

[Sit*20] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. “Implicit Neural Representations with Periodic Activation Functions”. In: *Proc. NeurIPS*. 2020.

[SZW19] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. “Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[SRC01] Peter-Pike J Sloan, Charles F Rose III, and Michael F Cohen. “Shape by example”. In: *Proc. of Symposium on Interactive 3D graphics*. ACM. 2001, pp. 135–143.

[Spe*22] Georg Sperl, Rosa M. Sánchez-Banderas, Manwen Li, Chris Wojtan, and Miguel A. Otaduy. “Estimation of Yarn-Level Simulation Models for Production Fabrics”. In: *ACM Transactions on Graphics (TOG)* 41.4 (2022).

[SH07] Jonathan Starck and Adrian Hilton. “Surface Capture for Performance-Based Animation”. In: *IEEE Computer Graphics and Applications* 27.3 (2007), pp. 21–31.

[SE17] Russell Stewart and Stefano Ermon. “Label-Free Supervision of Neural Networks with Physics and Domain Knowledge”. In: *Proc. of the AAAI Conference on Artificial Intelligence*. 2017, pp. 2576–2582.

[Sto*10] Carsten Stoll, Juergen Gall, Edilson De Aguiar, Sebastian Thrun, and Christian Theobalt. “Video-based reconstruction of animatable human characters”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 29.6 (2010).

[Su*21] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. “A-NeRF: Articulated Neural Radiance Fields for Learning Human Shape, Appearance, and Pose”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021.

[Su*20] Zhuo Su, Lan Xu, Zerong Zheng, et al. “RobustFusion: Human Volumetric Capture with Data-Driven Visual Cues Using a RGBD Camera”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2020, pp. 246–264.

[Suc*21] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. “iMAP: Implicit Mapping and Positioning in Real-Time”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021, pp. 6229–6238.

[Tan*20] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, et al. “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[Tan*16] Min Tang, Huamin Wang, Le Tang, Ruofeng Tong, and Dinesh Manocha. “CAMA: Contact-Aware Matrix Assembly with Unified Collision Handling for GPU-based Cloth Simulation”. In: *Computer Graphics Forum* 35.2 (2016).

[Tan*18] Min Tang, Tongtong Wang, Zhongyuan Liu, Ruofeng Tong, and Dinesh Manocha. “I-Cloth: Incremental Collision Handling for GPU-Based Interactive Cloth Simulation”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 37.6 (2018).

[Tau95] Gabriel Taubin. “A Signal Processing Approach to Fair Surface Design”. In: *Proc. of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH ’95. New York, NY, USA: Association for Computing Machinery, 1995, pp. 351–358.

[Tiw*20] Garvita Tiwari, Bharat Lal Bhatnagar, Tony Tung, and Gerard Pons-Moll. “SIZER: A Dataset and Model for Parsing 3D Clothing and Learning Size Sensitive 3D Clothing”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2020.

[Tiw*21] Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. “Neural-GIF: Neural Generalized Implicit Functions for Animating People in Clothing”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[Tom*17] Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. “Accelerating Eulerian Fluid Simulation With Convolutional Networks”. In: *International Conference on Machine Learning (ICML)*. 2017, pp. 3424–3433.

[TMB14] Aggeliki Tsoli, Naureen Mahmood, and Michael J. Black. “Breathing Life into Shape: Capturing, Modeling and Animating 3D Human Breathing”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 33.4 (2014), 52:1–52:11.

[Var*17] Gül Varol, Javier Romero, Xavier Martin, et al. “Learning from Synthetic Humans”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2017.

[Vid*20] Raquel Vidaurre, Igor Santesteban, Elena Garces, and Dan Casas. “Fully Convolutional Graph Neural Networks for Parametric Virtual Try-On”. In: *Computer Graphics Forum (Proc. SCA)* 39.8 (2020).

[Vla*08] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. “Articulated Mesh Animation from Multi-View Silhouettes”. In: *Proc. of ACM SIGGRAPH*. 2008, pp. 1–9.

[Vla*09] Daniel Vlasic, Pieter Peers, Ilya Baran, et al. “Dynamic Shape Capture Using Multi-View Photometric Stereo”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*. SIGGRAPH Asia ’09. Yokohama, Japan: Association for Computing Machinery, 2009.

[Wan18] Huamin Wang. “Rule-Free Sewing Pattern Adjustment with Precision and Efficiency”. In: *ACM Transactions on Graphics* 37.4 (July 2018).

[Wan*10] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James O’Brien. “Example-based wrinkle synthesis for clothing animation”. In: *ACM Transactions on Graphics* 29.4 (2010), p. 1.

[WOR11] Huamin Wang, James F O'Brien, and Ravi Ramamoorthi. "Data-Driven Elastic Models for Cloth: Modeling and Measurement". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 30.4 (2011), pp. 1–12.

[Wan*20] Jiahang Wang, Tong Sha, Wei Zhang, Zhoujun Li, and Tao Mei. "Down to the Last Detail: Virtual Try-on with Fine-Grained Details". In: *Proceedings of the 28th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 466–474.

[WGT21] Shaofei Wang, Andreas Geiger, and Siyu Tang. "Locally Aware Piecewise Transformation Fields for 3D Human Mesh Registration". In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2021.

[Wan*21] Shaofei Wang, Marko Mihajlovic, Qianli Ma, Andreas Geiger, and Siyu Tang. "MetaAvatar: Learning Animatable Clothed Human Models from Few Depth Images". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021.

[Wan*18] Tuanfeng Y Wang, Duygu Ceylan, Jovan Popović, and Niloy J Mitra. "Learning a Shared Shape Space for Multimodal Garment Design". In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 37.6 (2018).

[Wan*19] Tuanfeng Y Wang, Tianjia Shao, Kai Fu, and Niloy J Mitra. "Learning an Intrinsic Garment Space for Interactive Authoring of Garment Animation". In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 38.6 (2019).

[Wen04] Holger Wendland. *Scattered data approximation*. Vol. 17. Cambridge university press, 2004.

[WCF07] Ryan White, Keenan Crane, and D. A. Forsyth. "Capturing and Animating Occluded Cloth". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 26 (3 2007).

[Wol*21] Katja Wolff, Philipp Herholz, Verena Ziegler, et al. "3D Custom Fit Garment Design with Body Movement". In: *arXiv preprint arXiv:2102.05462* (2021).

[Wu*16] Jiajun Wu, Joseph J Lim, Hongyi Zhang, Joshua B Tenenbaum, and William T Freeman. "Physics 101: Learning Physical Object Properties from Unlabeled Videos". In: *The British Machine Vision Conference (BMVC)*. 2016.

[Wu*19] Zhonghua Wu, Guosheng Lin, Qingyi Tao, and Jianfei Cai. "M2E-Try On Net: Fashion from Model to Everyone". In: *Proc. of ACM International Conference on Multimedia*. 2019.

[Xia*20] Donglai Xiang, Fabian Prada, Chenglei Wu, and Jessica Hodgins. "MonoClothCap: Towards Temporally Coherent Clothing Capture from Monocular RGB Video". In: *2020 International Conference on 3D Vision (3DV)*. IEEE. 2020, pp. 322–332.

[Xie*22] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, et al. "Neural Fields in Visual Computing and Beyond". In: *Computer Graphics Forum* (2022).

[Xie*18] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. "tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 37.4 (2018).

[XAS21] Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. “H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021.

[XB16] Hongyi Xu and Jernej Barbič. “Pose-space subspace dynamics”. In: *ACM Trans. Graphics* 35.4 (2016), 35:1–35:14.

[Xu*14] Weiwei Xu, Nobuyuki Umentani, Qianwen Chao, et al. “Sensitivity-optimized Rigging for Example-based Real-Time Clothing Synthesis”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 33.4 (2014).

[Yan*22] Gengshan Yang, Minh Vo, Neverova Natalia, et al. “BANMo: Building Animatable 3D Neural Models from Many Casual Videos”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2022.

[Yan*21a] Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. “Geometry Processing with Neural Fields”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021.

[Yan*20] Han Yang, Ruimao Zhang, Xiaobao Guo, et al. “Towards Photo-Realistic Virtual Try-On by Adaptively Generating-Preserving Image Content”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[Yan*18] Jinlong Yang, Jean-Sébastien Franco, Franck Hétroy-Wheeler, and Stefanie Wuhrer. “Analyzing Clothing Layer Deformation Statistics of 3D Human Motions”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2018.

[YLL17] Shan Yang, Junbang Liang, and Ming C. Lin. “Learning-Based Cloth Material Recovery From Video”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2017.

[Yan*14] Yipin Yang, Yao Yu, Yu Zhou, et al. “Semantic parametric reshaping of human body models”. In: *Proc. of International Conference on 3D Vision (3DV)*. Vol. 2. IEEE. 2014, pp. 41–48.

[Yan*21b] Ze Yang, Shenlong Wang, Siva Manivasagam, et al. “S3: Neural Shape, Skeleton, and Skinning Fields for 3D Human Modeling”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2021.

[YWX19] Ruiyun Yu, Xiaoqi Wang, and Xiaohui Xie. “VTNFP: An Image-Based Virtual Try-On Network With Body and Clothing Feature Preservation”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. Oct. 2019.

[Yu*21] Tao Yu, Zerong Zheng, Kaiwen Guo, et al. “Function4D: Real-time Human Volumetric Capture from Very Sparse Consumer RGBD Sensors”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2021.

[Zes*22] Ryan S. Zesch, Bethany R. Witemeyer, Ziyan Xiong, David I. W. Levin, and Shinjiro Sueda. “Neural Collision Detection for Deformable Objects”. In: *arXiv preprint arXiv:2202.02309*. 2022.

[Zha*17] Chao Zhang, Sergi Pujades, Michael J Black, and Gerard Pons-Moll. “Detailed, Accurate, Human Shape Estimation from Clothed 3D Scan Sequences”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4191–4200.

[Zha*21a] Meng Zhang, Tuanfeng Y. Wang, Duygu Ceylan, and Niloy J. Mitra. “Dynamic Neural Garments”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 40.6 (2021).

[Zha*21b] Fuwei Zhao, Zhenyu Xie, Michael C. Kampffmeyer, et al. “M3D-VTON: A Monocular-to-3D Virtual Try-On Network”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2021.

[Zhe*21] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. “PaMIR: Parametric Model-Conditioned Implicit Representation for Image-based Human Reconstruction”. In: *IEEE Transactions on Pattern Analysis and Machine Intelegence* (2021).

[Zho*12] Zhenglong Zhou, Bo Shu, Shaojie Zhuo, et al. “Image-based clothes animation for virtual fitting”. In: *SIGGRAPH Asia 2012 Technical Briefs*. ACM. 2012, p. 33.

[Zhu*20] Heming Zhu, Yu Cao, Hang Jin, et al. “Deep Fashion3D: A Dataset and Benchmark for 3D Garment Reconstruction from Single Images”. In: *Proc. of European Conference on Computer Vision (ECCV)*. 2020.

[Zhu*17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks ”. In: *Proc. of IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2223–2232.

[Zhu*19] Yinhao Zhu, Nicholas Zabaras, Phaedon-Stelios Koutsourelakis, and Paris Perdikaris. “Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data”. In: *Journal of Computational Physics* 394 (2019), pp. 56–81.

[ZB15] Silvia Zuffi and Michael J Black. “The stitched puppet: A graphical model of 3D human shape and pose”. In: *Proc. of Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3537–3546.

[ZBO13] Javier S Zurdo, Juan P Brito, and Miguel A Otaduy. “Animating Wrinkles by Example on Non-Skinned Cloth”. In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 19.1 (2013), pp. 149–158.

Resumen

A

La ropa desempeña un papel fundamental en nuestra vida cotidiana. Cuando elegimos qué ropa comprar o vestir, guiamos nuestra decisión en base al estilo y ajuste de las prendas. Por esta razón, la mayor parte de la ropa se compra en tiendas físicas, tras comprobar en un probador cómo luce la ropa en nuestro propio cuerpo. La madurez del campo de los gráficos por computador y de la simulación de textiles ofrece nuevas oportunidades para revolucionar el proceso de compra de ropa mediante el desarrollo de *probadores virtuales*. Sin embargo, los prototipos de probadores virtuales existentes hasta la fecha carecen de la precisión e interactividad requerida para competir con los probadores de ropa físicos.

El objetivo de esta tesis es desarrollar nuevos métodos para predecir el ajuste de una prenda que satisfagan los exigentes requisitos de precisión, interactividad y escalabilidad necesarios para proporcionar una buena experiencia. Para ello, proponemos nuevos modelos basados en aprendizaje automático que permiten generar animaciones realistas de avatares y ropa 3D con un coste computacional mucho menor que el de los métodos tradicionales basados en física. A lo largo de la tesis también abordamos limitaciones habituales de los métodos basados datos, concretamente, proponemos novedosos mecanismos de autosupervisión que garantizan el cumplimiento de restricciones físicas y reducen la dependencia respecto a los datos de entrenamiento. Finalmente, también proponemos un método eficiente para resolver colisiones entre capas de ropa que permite combinar distintas prendas y comprobar, al instante, cómo se ajustan al cuerpo del usuario.

Figure A.1: Ejemplo de las deformaciones de ropa que podemos generar, en cuestión de milisegundos, gracias a los métodos desarrollados en esta tesis.

A.1 Antecedentes

Los avatares y la ropa virtual son los pilares fundamentales de un probador de ropa virtual. A continuación presentamos los trabajos más relevantes en estos campos y sus respectivas limitaciones.

Avatares virtuales

El modelado del cuerpo humano ha sido objeto de investigación tanto en la comunidad de visión artificial como de los gráficos 3D. Las tecnologías existentes son capaces de producir representaciones virtuales precisas de una persona, pero la precisión de estos métodos está ligada al uso de costosos estudios multicámara y marcadores físicos [SH07; Vla*08; Vla*09; NH14]. Existen líneas de investigación en curso que buscan hacer este proceso más accesible mediante la generación de avatares a partir de una única imagen RGB [Sai*19; Sai*20; Zha*21b].

Si bien estos métodos permiten generar un avatar 3D específico para un usuario, en esta tesis centramos nuestra atención en los modelos humanos paramétricos, que son capaces de representar un amplio abanico de cuerpos utilizando una parametrización de baja dimensión. Concretamente, los métodos desarrollados en esta tesis utilizan el modelo SMPL [Lop*15], un modelo humano paramétrico construido a partir de datos de personas reales. Este modelo es ampliamente utilizado por la comunidad científica para abordar problemas relacionados con avatares virtuales, como la estimación de parámetros de cuerpo (*e.g.*, pose, forma) que mejor se ajustan a la foto de una determinada persona [Bog*16; Kan*18; Omr*18; Pav*19; Fen*21].

Además de tener un modelo preciso del cuerpo del humano, también es importante modelar cómo se deforma el cuerpo en movimiento. Ello implica el desarrollo de métodos capaces de predecir las deformaciones del tejido blando que induce un determinado movimiento. Una manera de simular este tipo de deformaciones es mediante el uso de modelos físicos [Cap*02; LCA05; Liu*13; XB16; Pai*18; Rom*20], pero estos modelos conllevan un elevado coste computacional y requieren un complejo proceso de estimación de parámetros mecánicos.

La alternativa a los modelos basados en física son los modelos basados en datos. El desarrollo de tecnologías de escaneo 3D/4D [Bra*08; CBI10; Dou*15; Bog*17; Rob*17; Pon*17] ha posibilitado la captura y reconstrucción de secuencias dinámicas de actores en

movimiento con gran nivel de detalle. Esto ha dado lugar a conjuntos de datos de alta calidad que posteriormente han sido utilizados para construir, mediante técnicas de aprendizaje automático, modelos de regresión que predicen deformaciones de tejido blando en función del movimiento y la forma del cuerpo [Pon*15; Lop*15; CO18]. En el Capítulo 3 proponemos un nuevo modelo basado en aprendizaje que mejora el detalle de las deformaciones generadas y tiene una capacidad de generalización superior a la de los métodos existentes.

Ropa virtual

Actualmente existen múltiples herramientas para el diseño de ropa virtual (*e.g.*, Optitex, Marvelous Designer), la digitación de prendas reales [Sch*05; WCF07; Bra*08; Pon*17], el ajuste automático de patrones [Bar*16; Wan18], e incluso métodos para crear ropa a partir de esbozos [Li*17a; Wan*18]. A pesar de abordar problemas muy distintos, todas estas herramientas requieren estimar deformaciones de telas para realizar su función. Los métodos existentes para modelar deformaciones de tela se pueden categorizar en dos grupos: modelos basados en física y modelos basados en datos.

Los modelos basados en física utilizan discretizaciones de modelos de mecánica clásica para predecir cómo se deforma la tela [Nea*06]. Estos métodos producen simulaciones altamente realistas, generalizan a múltiples prendas, y pueden gestionar colisiones entre la ropa y el cuerpo. Sin embargo, dado su alto coste computacional, no proporcionan la combinación de precisión y eficiencia requeridos para el desarrollo de un probador de ropa virtual. El diseño de simuladores de tela más eficientes es una línea de investigación activa [Ben*14; Bou*14; Ly*20; Lee*10; NSO12; Tan*16; FTP16; Tan*18].

Los modelos basados en datos predicen la deformación de una tela a partir de datos precomputados [Kim*13] o regresores aprendidos mediante técnicas de aprendizaje automático [Gua*12]. Trabajos recientes en este campo utilizan redes neuronales para predecir deformaciones de ropa en función de la pose del cuerpo [LCT18; Wan*19], la forma del cuerpo [Vid*20], la pose y la forma [SOC19; BME20], parámetros de diseño [PLP20; Wan*18; Ma*20], o incluso el tamaño de la prenda [Tiw*20]. Los modelos de deformación de ropa desarrollados en esta tesis (Capítulos 4, 5) pertenecen a esta línea de investigación, pero también incorporan conocimiento de modelos físicos para mejorar la calidad de los resultados y reducir la dependencia en datos. Además, en esta tesis también proponemos métodos novedosos para gestionar el contacto con el cuerpo (Capítulo 6) y entre capas de ropa (Capítulo 7), solventando así algunas de las limitaciones más importantes de los métodos basados en datos.

A.2 Objetivos

El principal objetivo de esta tesis es el desarrollo de probadores de ropa virtuales, lo cual a su vez implica el desarrollo de modelos eficientes y precisos de avatares y ropa 3D. En su forma más simple, un probador de ropa de virtual implica estos pasos: primero, el usuario proporciona información de su cuerpo (*e.g.*, imágenes o medidas) y selecciona un conjunto de prendas, después, la aplicación de prueba de ropa predice cómo se ajustan esas prendas al usuario, y finalmente, se muestra el resultado. Si bien el concepto de los probadores virtuales no es novedoso, los prototipos existentes se ven severamente limitados por problemas de alta complejidad técnica. Para poder comprender bien estos problemas, es necesario definir primero cuáles son las propiedades que cualquier probador virtual debiera satisfacer:

- **Precisión.** Para ser útil, el probador virtual debe proporcionar estimaciones precisas de cómo se ajusta la ropa al cuerpo del usuario. El sistema también tiene que ser preciso a la hora de transmitir el estilo de la prenda y las propiedades visuales de su tejido.

- **Interactividad.** Para ser usable, el probador virtual debe proporcionar resultados con un retraso mínimo y permitir al usuario probar combinaciones de prendas de forma interactiva. Modos adicionales de interacción (*e.g.*, mediante avatares en movimiento) también pueden contribuir positivamente a la usabilidad de la herramienta.

- **Escalabilidad.** Para ser rentable, el coste de predecir los resultados y el esfuerzo de añadir nuevas prendas al sistema deben ser lo más bajos posible. El sistema también tiene que dar soporte a una amplia gama de cuerpos y a una combinación casi ilimitada de prendas.

Actualmente no existen métodos que satisfagan estos tres requisitos simultáneamente. Por ejemplo, los métodos basados en física [KJM08; Sel*09; NSO12; Cir*14] pueden predecir el ajuste de una prenda con alta precisión, pero su elevado coste computacional implica sacrificar interactividad (el usuario tiene que esperar para ver el resultado) y escalabilidad (la simulación tiene un coste significativo por usuario). Por otro lado, los métodos basados en imagen [SM06; Zho*12; HSR13; HFE13; Han*18; CML21] sintetizan imágenes del usuario vistiendo la ropa seleccionada, pero carecen de precisión a la hora de predecir el ajuste de las prendas. Además, los métodos basados en imagen dependen en gran medida de fotografías de modelos profesionales, lo cual introduce un sesgo hacia formas corporales que no son representativas de toda la población.

El objetivo de esta tesis es, por tanto, desarrollar nuevos métodos para probadores virtuales que sean precisos, interactivos y escalables. Para ello, usamos como punto de partida la literatura existente sobre avatares y ropa virtual, y exploramos el uso de técnicas de aprendizaje automático para diseñar modelos que solventen las limitaciones de los métodos actuales.

A.3 Metodología

En esta sección describimos la metodología seguida durante el desarrollo de esta tesis.

Revisión bibliográfica

Una vez definidos los objetivos de la tesis, se procedió a hacer una revisión bibliográfica para identificar líneas de investigación prometedoras. De esta revisión bibliográfica inicial surge nuestra apuesta por los modelos de ropa y cuerpos basados en aprendizaje, que al comienzo de esta tesis eran prácticamente inexistentes. En la actualidad el uso de técnicas de aprendizaje automático para modelar objetos deformables se ha convertido en una línea de investigación propia, con un flujo de nuevos trabajos e ideas cada vez mayor. Por ello, ha sido fundamental tratar la revisión bibliográfica como un documento vivo en el que incorporar los avances del campo a medida que este avanzaba. Esto nos ha permitido también identificar nuevas tendencias, como el uso de redes neuronales para representar superficies implícitas, que han resultado ser claves para el desarrollo de esta tesis.

Diseño de modelos eficientes de deformación de tejido blando

Los modelos existentes de deformación de tejido blando [Pon*15; Lop*15; CO18] consiguen resultados prometedores pero tienen una capacidad de generalización limitada, lo cual se traduce en dinámicas de tejido blando altamente amortiguadas. Un aspecto clave para solventar esta limitación fue identificar la raíz de este problema: los datos. Dada la complejidad de adquirir datos reales de deformaciones de tejido blando, los métodos existentes utilizan un conjunto de datos reducido que consiste en unos pocos sujetos realizando movimientos similares. Como cada sujeto tiende a realizar los movimientos de forma ligeramente distinta (consecuencia de su anatomía o su estilo personal), esto introduce una variabilidad en los datos que los modelos basados en aprendizaje no son capaces de interpre-

tar correctamente. Nuestra contribución principal en este ámbito se centra en normalizar las secuencias de cada sujeto y eliminar esta variabilidad de los datos. El Capítulo 3 demuestra que de este modo se pueden entrenar modelos con una alta capacidad de generalización sin necesidad de usar más datos. Durante el desarrollo de este proyecto también exploramos el uso de distintas arquitecturas de redes neuronales, lo que nos permitió diseñar un modelo que mejora significativamente el nivel de detalle de las deformaciones de tejido blando generadas.

Diseño de modelos eficientes de deformación de ropa

Los modelos de deformación de ropa basados en datos previos a esta tesis [De *10; Gua*12; LCT18] estaban entrenados para un único sujeto o no modelaban de manera realista la deformación una prenda al ser vestida por distintos cuerpos. Tampoco existían buenos conjuntos de datos para abordar este problema. Por ello, el primer paso fue adaptar un simulador de telas existente [NSO12] y generar un conjunto de datos simulados para una prenda vestida por diversos cuerpos en movimiento. Esto nos permitió desarrollar el modelo presentado en el Capítulo 4, que fue el primer modelo basado en aprendizaje capaz de generar deformaciones dinámicas de una prenda y generalizar de forma precisa a distintos cuerpos. Este método ha constituido un avance significativo de cara al desarrollo de probadores de ropa precisos e interactivos, pero el elevado coste de generar los datos de entrenamiento hace que este método no sea escalable.

Esta limitación nos ha llevado a explorar nuevos métodos de entrenamiento que no requieren el uso de datos precomputados. El trabajo realizado en este ámbito ha dado lugar al método que se presenta en el Capítulo 5, que elimina completamente la necesidad de precomputar datos y reduce los tiempos de entrenamiento de cada prenda de $\sim$200h a únicamente 2h.

Diseño de métodos para resolver contacto entre superficies

Uno de los mayores retos de los modelos de ropa (tanto los basados en física como los basados en datos) es gestionar el contacto entre superficies. Dada la complejidad técnica del problema, decidimos abordar primero el contacto entre una prenda y el cuerpo, y posteriormente el contacto entre varias prendas. El Capítulo 6 presenta un método que resuelve las colisiones de una prenda respecto al cuerpo en tiempo de entrenamiento, de forma que las deformaciones de ropa producidas en tiempo de ejecución estén libres de colisiones. El Capítulo 5 presenta un método que resuelve colisiones entre múltiples capas

de ropa en cuestión de milisegundos, permitiendo así que el usuario combine prendas de manera interactiva. Previo al desarrollo de estos dos métodos, no era posible combinar múltiples prendas 3D de forma interactiva y totalmente automática.

A.4 Resultados

Estas son las principales contribuciones de la tesis:

- Un método basado en aprendizaje para el modelado de dinámica de tejido blando en función de la forma y movimiento del cuerpo. Este método se sustenta en tres contribuciones clave que nos permiten modelar dinámicas altamente realistas y lograr una mejor capacidad de generalización que la de los métodos existentes. En primer lugar, proponemos un novedoso descriptor de movimiento que mejora la representación de pose estándar y elimina características específicas de cada sujeto; en segundo lugar, un regresor recurrente basado en redes neuronales que generaliza a movimientos y formas de cuerpos no vistos; y en tercer lugar, un subespacio de deformación no-lineal altamente eficiente capaz de representar deformaciones de tejido blando de todo tipo de cuerpos. (Capítulo 3)

- Un método basado en aprendizaje para producir deformaciones detalladas de ropa en cuestión de milisegundos. Nuestro método se apoya en técnicas de *skinning* tradicionales para obtener una aproximación inicial del movimiento de la prenda. A continuación, mejoramos este modelo aproximado introduciendo un vector de desplazamientos correctivos calculados por una red neuronal recurrente. Con el objetivo de obtener animaciones de ropa realistas, la red aprende estos desplazamientos a partir de secuencias obtenidas mediante simulación física, y es capaz de generalizar a avatares con cuerpos distintos. (Capítulo 4)

- Un método autosupervisado para aprender deformaciones de ropa sin necesidad de datos de entrenamiento. Este método surge de la observación de que los modelos de deformación basados en la física, que tradicionalmente se resuelven fotograma a fotograma mediante integradores implícitos, pueden reformularse como un problema de optimización. Aprovechamos este esquema basado en optimización para formular un conjunto de funciones de pérdida basadas en física que pueden utilizarse para entrenar redes neuronales sin necesidad de un conjunto de datos. Esto nos permite aprender modelos interactivos para prendas con dinámica y alto nivel de detalle, y conseguir tiempos de entrenamiento significativamente menores en comparación a los métodos supervisados. (Capítulo 5)

- Un subespacio generativo de deformaciones de ropa que nos permite aprender, por primera vez, un modelo que aborda eficazmente las colisiones entre la ropa y el cuerpo. A diferencia de los métodos existentes, que requieren un indeseable postproceso para arreglar las interpenetraciones entre tela y cuerpo, nuestro enfoque produce directamente resultados que no colisionan con el cuerpo. La clave de nuestro éxito es un nuevo espacio canónico para representar prendas que elimina las deformaciones inducidas por la pose y la forma del cuerpo. Para ello, presentamos un nuevo modelo difuso del cuerpo humano, que extrapola las propiedades de la superficie del cuerpo a cualquier punto 3D. Aprovechamos esta representación para entrenar un subespacio generativo de deformaciones con un novedoso término de colisión autosupervisado que aprende a resolver, de forma fiable, las colisiones entre la ropa y el cuerpo. (Capítulo 6)

- Un método novedoso para gestionar colisiones entre capas de ropa que permite combinar prendas de forma interactiva. Para ello, representamos las prendas de manera implícita utilizando campos neuronales y separamos estos campos para obtener superficies libres de colisiones. El ingrediente clave es un operador de proyección neuronal que se aplica directamente en los campos, no en las representaciones explícitas de la superficie, y nos permite separar las capas de ropa eficientemente. (Capítulo 7)

Estas contribuciones han dado lugar a las siguientes publicaciones:

- Igor Santesteban, Miguel A. Otaduy, and Dan Casas. "Learning-Based Animation of Clothing for Virtual Try-On". *Computer Graphics Forum (Proc. Eurographics)* (2019)

- Igor Santesteban, Elena Garces, Miguel A. Otaduy, and Dan Casas. "SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Humans". *Computer Graphics Forum (Proc. Eurographics)* (2020)

- Igor Santesteban, Nils Thuerey, Miguel A. Otaduy, and Dan Casas. "Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On". *Proc. of Computer Vision and Pattern Recognition (CVPR)* (2021)

- Igor Santesteban, Miguel A. Otaduy, and Dan Casas. "SNUG: Self-Supervised Neural Dynamic Garments". *Proc. of Computer Vision and Pattern Recognition (CVPR)* (2022)

- Igor Santesteban, Nils Thuerey, Miguel A. Otaduy, and Dan Casas. "ULNeF: Untangled Layered Neural Fields for Mix-and-Match Virtual Try-On". *En revisión.*

A.5 Conclusiones

A lo largo de esta tesis hemos abordado importantes retos en el campo de los avatares y la ropa virtual; desde la estimación de la dinámica del tejido blando del cuerpo (Capítulo 3) y la deformación de ropa (Capítulos 4, 5), hasta la gestión del contacto entre prendas y cuerpos (Capítulo 6) y entre múltiples capas de ropa (Capítulo 7). Consideramos que esta tesis ha supuesto un avance significativo hacia el desarrollo de probadores de ropa virtuales precisos, interactivos y escalables. Gracias a los métodos desarrollados, hemos implementado una aplicación interactiva que sirve de demostrador de nuestras ideas (Figura A.2). Si bien la implementación actual está limitada a una pose estática, confiamos en seguir ampliando su alcance incorporando nuestro trabajo sobre avatares animados y ropa en movimiento.

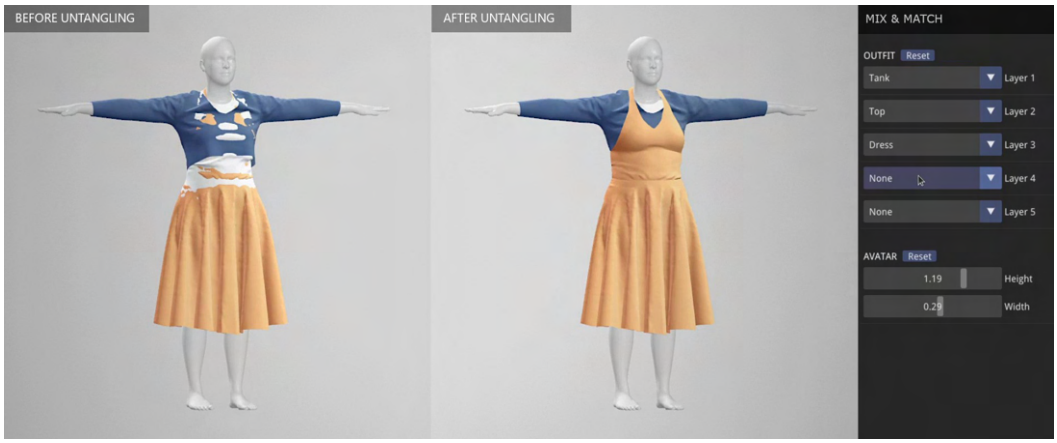


Figure A.2: Captura de pantalla de nuestra aplicación interactiva. Esta aplicación conlleva importantes retos técnicos que los métodos del estado del arte (izquierda) no son capaces de resolver. Nuestros modelos de ropa y contacto (derecha), son capaces de gestionar casos de alta complejidad en cuestión de milisegundos.

En términos generales, en estos últimos años hemos sido testigos de enormes avances en el ámbito de los avatares y ropa 3D. Somos optimistas de que más pronto que tarde las tecnologías de prueba de ropa virtual estarán listas para el público general. Además de desarrollar métodos que sean precisos y fiables, también hemos invertido gran esfuerzo en desarrollar métodos que tengan en cuenta la diversidad del cuerpo humano. Una diversidad que no siempre es reconocida por una industria que tiende a favorecer estándares de belleza que no representan a la sociedad en su conjunto. También esperamos que el desarrollo de herramientas digitales para la industria de la moda sea un punto de inflexión para hacer frente al grave impacto medioambiental de esta industria.