Latent Disentanglement for the Analysis and Generation of Digital Human Shapes

Simone Foti

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Medical Physics and Biomedical Engineering
University College London

June 30, 2023

I, Simone Foti, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Acknowledgements

I express my sincere gratitude to all those who have helped me throughout the process of completing this thesis. Despite having written several theses before, I have never written an acknowledgement section because I was waiting until the end of the PhD, *dulcis in fundō!* Although I will do my best to keep the acknowledgements brief, it might take a while.

First and foremost, I would like to thank my primary supervisor **Prof. Matt Clarkson** for his help and support. Thank you for your advice, which was valuable both in science and life. Most of all, thank you for giving me the freedom to pursue research topics that I found interesting.

I would like to extend my sincere thanks to my second supervisor, **Prof. Danail Stoyanov**, for his feedback and for always being available when needed.

A special thanks goes to **Dr. Bongjin Koo**, friend and invaluable supervisor. You have always been a valuable sounding board and a source of constructive criticism, pushing me to question assumptions and challenge myself. Even though you may not have realised it, your feedback and advice have been a critical part of shaping my research and have helped me to grow as a scholar.

I would also like to thank all the clinical collaborators I had the pleasure to work with. In particular, I would like to thank **Prof. David J. Dunaway**, **Prof. Silvia Schievano**, **N. u. Owase Jeelani**, and **Alexander J. Rickart** from the UCL FaceValue Craniofacial Research Unit as well as **Moustafa Allam** and **Prof. Brian Davidson** from the SmartLiver team for their

insights on the clinical and translational aspects of this thesis.

I am also incredibly thankful to all the people who supervised or worked with me during my internships at Disney Research Studios and Adobe Research. Although this thesis does not contain any works conducted during these internships, both experiences inspired my research directions and allowed me to grow on a professional level. I would particularly like to thank **Derek Bradley** and all the members of the Digital Humans group at Disney Research Studios who contributed to my project: **Paulo F. U. Gotardo, Gaspard Zoss, and Prashanth Chandran**. From Adobe Research, I would like to thank **Jérémy Levallois, Valentin Deschaintre,** and **Robin Faury**. I am deeply grateful for the opportunity to have worked alongside such remarkable groups of talented and inspiring researchers. Each and every one of you has played a significant role in influencing my research, making me even more passionate about computer graphics and digital humans.

During the PhD I met a lot of great people that I consider to be my friends, arguably far too many to mention here. I will try to be brief! I would like to start by thanking **Bongjin** (yes, you get acknowledged twice!), as well as **João**, and **Henry**, with whom I had a lot of fun at work, at the pub, at dinner, and most importantly at the table tennis table. Table tennis surely played a crucial role in creating a sense of social cohesion and community within our research centre. In fact, I would like to extend my gratitude also to **Nick**, **Matas**, and all the other dear colleagues who played with me during our breaks from work. Thank you to **Alex** and **Luke** –also known as the basement guys of the FaceValue group– with whom I had great fun in the last year. Thank you to my fellow intern **Sebastian** for the great company during our countless late nights at Disney Research. Many thanks to **Julien**, whom I had the pleasure of meeting during my internship at Adobe Research, for the enjoyable dinners, insightful discussions, and shared Whiskey. Thanks to **Bea** and **François** for the delightful walks we took together at Hampstead Heath, and for the many insightful research discussions we had during those

walks. Grazie mille for the fun we had together also to Luca M. and the way too many Italians of Weiss who prevented me from forgetting my native language. Grazie mille also to Gabri and Luca C., whom I had the pleasure of meeting during my MSc while visiting Carnegie Mellon University. Thank you for the wonderful memories we created together while travelling around the US, taking overnight breaks from work, cooking delicious Italian meals, and simply enjoying each other's company. More importantly, thank you for being a great inspiration for me to pursue a PhD!

Now, I would like to take a moment to express my gratitude to some special people whom I have known since long before beginning my PhD. To those who may not understand Italian, my apologies. I am con dent that with the help of an automatic translator or some good-natured gesturing, you will be able to follow along.

Azioni, decisioni, esperienze, amicizie e conoscenze in uenzano inevitabilmente lo sviluppo della persona già a partire dalla nascita. Con questo presupposto dovrei scrivere una lista di ringraziamenti pressoché in nita. Non essendo né possibile né quantomeno sensato, limiterò i ringraziamenti alle persone che hanno maggiormente in uenzato la mia vita e con le quali sono rimasto maggiormente in contatto. Ci tengo a sottolineare che vorrei ringraziare molte più persone e, nonostante i loro nomi non siano esplicitamente stati menzionati, erano impressi nella mia mente durante la stesura di questa sezione.

Ritengo che un ringraziamento speciale sia dovuto a due dei miei più dati amici: Davide e Luca. Ringrazio Davide, mio compagno di barca, per tutte le giornate di allenamento, di vacanza, di gioco e divertimento, nonché per le discussioni più strampalate ed interessanti che abbia mai avuto al di fuori di un contesto lavorativo. Ringrazio Luca S. per aver reso divertenti anche le più noiose giornate scolastiche, per tutti gli indimenticabili momenti passati assieme e la sue invidiabili ambizioni personali e lavorative, che hanno fortemente condizionato e motivato anche me. Nonostante la lontananza abbia contribuito a farci perdere un po' i contatti, ringrazio anche Lorenzo e Giulia,

i quali hanno fortemente contribuito alla persona che sono.

Uno dei ringraziamenti più speciali non può che essere rivolto a Sara, la quale è sempre stata al mio fianco seguendomi in giro per il mondo. Grazie per esserci sempre e (praticamente) da sempre! Grazie per sopportarmi nonostante io spesso viva su quello che sembra essere un fuso orario completamente diverso. Ma sopratutto, grazie per il tuo appoggio incondizionato.

Un altro ringraziamento speciale va indubbiamente alla mia famiglia, che mi ha sempre supportato ed incoraggiato. Ritengo non ci sia membro famigliare che non meriti un ringraziamento, quindi estendo la mia gratitudine a tutta la famiglia. Inoltre ritengo siano dovute anche delle scuse a mia Mamma, ed ai miei fratelli Mario e Valeria, ai quali non sono stato vicino come avrei sperato negli ultimi anni. In fine, ci tengo a ringraziare mia nonna Alida, la quale fu anche la mia prima maestra che, con infinita pazienza, mi insegnò a disegnare, leggere e scrivere ben prima di iniziare la scuola. Questa tesi la dedico a te sebbene tu non sia qui per poterla leggere.

Abstract

Analysing and generating digital human shapes is crucial for a wide variety of applications ranging from movie production to healthcare. The most common approaches for the analysis and generation of digital human shapes involve the creation of statistical shape models. At the heart of these techniques is the de nition of a mapping between shapes and a low-dimensional representation. However, making these representations interpretable is still an open challenge. This thesis explores latent disentanglement as a powerful technique to make the latent space of geometric deep learning based statistical shape models more structured and interpretable. In particular, it introduces two novel techniques to disentangle the latent representation of variational autoencoders and generative adversarial networks with respect to the local shape attributes characterising the identity of the generated body and head meshes. This work was inspired by a shape completion framework that was proposed as a viable alternative to intraoperative registration in minimally invasive surgery of the liver.
In addition, one of these methods for latent disentanglement was also applied to plastic surgery, where it was shown to improve the diagnosis of craniofacial syndromes and aid surgical planning.

Impact Statement

The latent disentanglement techniques introduced to control the generation of digital human shapes has immediate application in the creative industry. In fact, the proposed techniques can simplify the character generation process in movie and video game production, augmented reality, and virtual reality. In these elds, highly skilled artists usually engage in months-long manual design processes to create realistic digital human shapes. The proposed generative models can generate a wide variety of realistic human subjects in interactive interfaces that provide real-time and ne-grained control over the generation. The graphical user interface developed to demonstrate this (Ch. 6), highlights also how the proposed methods can democratise the character generation process. In fact, it could be embedded in video games and metaverse applications, where users can create their own characters without being limited by the design choices of the current platforms {that usually perform local linear interpolations between two artist-generated shapes, or let users select a shape from a set of prede ned examples.

The proposed models can e ectively be considered interpretable priors. As such, they can bene t a wide variety of computer vision applications. One future application could be the intraoperative liver surface completion framework that was tackled during the early part of this work. Even without local disentanglement, the framework showed its potential as an alternative to intraoperative surface registration. While more work is still needed, the use of latent disentanglement could provide a new level of control for intraoperative registration tasks, and enable computationally feasible real-time registration

methods with realistic priors on the modes of variation.

The use of latent disentanglement to analyse head shapes of patients with Apert, Crouzon, and Muenke syndromes provided insights into how di erent anatomical sub-units in uence syndromes. The integration of the method in the current clinical work ow can already improve the diagnosis of complex cases, assist on the selection of the surgical approach, and aid surgical planning by showing the statistically desired result. In addition, it also opened new avenues for the objective evaluation of surgical outcomes. Finally, if data were available, this method could have a signi cant diagnostic impact on the evaluation of syndromes with more subtle deformities or in prenatal stages.

List of Publications

1. Simone Foti, Bongjin Koo, Danail Stoyanov, Matthew J. Clarkson. 3D Generative Model Latent Disentanglement via Local Eigenprojection. Computer Graphics Forum 2023 [63]*.

2. Simone Foti, Alexander J. Rickart, Bongjin Koo, Eimear O'Sullivan, Lara S. van de Lande, Athanasios Papaioannou, Roman Khonsari, Danail Stoyanov, N. u. Owase Jeelani, Silvia Schievano, David J. Dunaway, Matthew J. Clarkson. Latent Disentanglement in Mesh Variational Autoencoders Improves the Diagnosis of Craniofacial Syndromes and Aids Surgical Planning. Computer Methods and Programs in Biomedicine 2023 (under review)*.

3. Simone Foti, Bongjin Koo, Danail Stoyanov, Matthew J. Clarkson. 3D Shape Variational Autoencoder Latent Disentanglement via Mini-Batch Feature Swapping for Bodies and Faces. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022 [62]*.

4. Simone Foti, Bongjin Koo, Thomas Dowrick, João Ramalhinho, Moustafa Allam, Brian Davidson, Danail Stoyanov, Matthew J. Clarkson. Intraoperative Liver Surface Completion with Graph Convolutional VAE. MICCAI { Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis, pp. 198-207. Springer, Cham, 2020 [61]*.

---

*The UCL Research Paper Declaration forms are available in Appendix A.2

Contents

1 Introduction 22

1.1 Problem Statement and Motivation 22

1.2 Aim of the Thesis 25

1.3 Structure of the Thesis 26

2 Background 28

2.1 Shape Representation 28

2.2 Statistical Shape Models 31

2.2.1 Principal Component Analysis 34

2.2.2 Variational Autoencoders 36

2.2.3 Generative Adversarial Networks 39

2.3 Geometric Deep Learning 41

2.3.1 Feature Steered Graph Convolutions 43

2.3.2 Spiral++ Convolutions 43

2.3.3 Pooling and Un-pooling with Fixed Topologies 44

2.4 Latent Disentanglement 45

2.4.1 -VAE 47

2.4.2 Disentangled Inferred Prior VAE 49

2.4.3 Factor VAE 50

2.5 Spectral Geometry 51

2.6 An Overall Perspective on the Topics 54

3 Intraoperative Shape Completion 55

3.1 Introduction 56
3.1.1 Computer Assisted and Minimally Invasive Surgery . . . 56
3.1.2 Shape Completion for Computer Assisted Liver Surgery . 58
3.2 Methods 59
3.2.1 Shape Generator 60
3.2.2 Shape Completion 63
3.3 Results 65
3.3.1 Qualitative Results 66
3.3.2 Quantitative Results 67
3.4 Discussion 68

4 Swap Disentangled Variational Autoencoder 70
4.1 Introduction 71
4.2 Method 72
4.2.1 Mesh Variational Autoencoder 73
4.2.2 Mini-Batch Attribute Swapping and Latent Consistency Loss 75
4.3 Experiments 77
4.3.1 Datasets 77
4.3.2 Implementation Details 78
4.3.3 Comparison with Other Methods 78
4.3.4 Evaluation of Latent Disentanglement 81
4.3.5 Ablation Study 86
4.3.6 Generalisation Capabilities 87
4.3.7 Latent Space Interpolation 88
4.3.8 Direct Manipulation 89
4.4 Discussion 90

5 Craniofacial shape analysis 92
5.1 Introduction 93
5.1.1 Virtual Surgical Planning in Plastic Surgery 93

5.1.2 Virtual Surgical Planning of Craniofacial Syndromes . . 95
5.2 Methods 99
5.2.1 Data sources and processing 99
5.2.2 Data augmentation with spectral interpolation 100
5.2.3 Swap disentangled 3D mesh variational autoencoder . . . 101
5.2.4 Latent manifold visualisation 102
5.2.5 Syndrome classi cation 104
5.2.6 Evaluating the e ects of di erent surgical approaches using latent interpolation 104
5.3 Results 106
5.3.1 Data augmentation 110
5.3.2 Model and latent disentanglement evaluation 111
5.3.3 Latent Manifold visualisation 112
5.3.4 Syndrome classi cation 113
5.3.5 Evaluating the e ects of di erent surgical approaches . . 115
5.3.6 Analysis of Complex Cases 119
5.4 Discussion 121

6 Local Eigenprojection Disentanglement 125
6.1 Introduction 126
6.2 Method 127
6.2.1 Local Eigenprojection Loss 128
6.2.2 Mesh Variational Autoencoder 131
6.2.3 Mesh Generative Adversarial Networks 132
6.3 Experiments 133
6.3.1 Datasets 133
6.3.2 Implementation Details 134
6.3.3 Local Eigenprojection Distributions 135
6.3.4 Comparison with Other Methods 137
6.3.5 Evaluation of Latent Disentanglement 141
6.3.6 LED-VAE Additional Experiments 146

6.4 Discussion 153

7 Conclusion 155

7.1 Limitations and Future Work 158

Appendices 162

A Appendix 162

A.1 Architectures 162

A.2 UCL Research Paper Declaration Forms 166

Bibliography 178

Acronyms

AE Autoencoder. 24

AI Arti cial Intelligence. 22

AR Augmented Reality. 22

CAS Computer Assisted Surgery. 22

CT Computed Tomography. 23

DIP-VAE Disentangled Inferred Prior VAE. 45

DL Deep Learning. 24

ELBO Evidence Lower Bound. 37

GAN Generative Adversarial Network. 24

GDL Geometric Deep Learning. 24

GUI Graphical User Interface. 126

KL Kullback{Leibler. 36

LDA Linear Discriminant Analysis. 98

LED Local Eigenprojection Disentangled. 126

LED-Model Local Eigenprojection Disentangled Model. 27

LSGAN Least Squares GAN. 132

MIS Minimally Invasive Surgery. 26

ML Machine Learning. 31

MRI Magnetic Resonance Imaging. 23

PCA Principal Component Analysis. 24

PSP Preoperative Surgical Planning. 93

QDA Quadratic Discriminant Analysis. 103

SD-VAE Swap Disentangled VAE. 26

SSM Statistical Shape Model. 28

SVD Singular Value Decomposition. 34

SVM Support Vector Machines. 97

TC Total Correlation. 51

VAE Variational Autoencoder. 24

VR Virtual Reality. 22

VSP Virtual Surgical Planning. 93

WGAN Wasserstein GAN. 132

List of Figures

2.1 Explicit shape representations. 29

2.2 Random samples generated by a bundle of PCA models trained each on a di erent shape attribute. 35

2.3 VAE architecture 36

2.4 GAN architecture 40

3.1 Schematic overview of the proposed intraoperative shape completion method. 60

3.2 Schematic description of the intraoperative liver shape completion. 64

3.3 Data augmentation and qualitative results of the proposed itraoperative shape completion method. 66

3.4 Quantitative results of the proposed itraoperative shape completion method. 67

4.1 Schematic description of the Swap Disentangled Variational Autoencoder (SD-VAE). 73

4.2 Examples of attribute swapping performed to achieve disentanglement with SD-VAE. 75

4.3 Random samples generated with SD-VAE and other state-of-the-art methods. 79

4.4 Random samples and visual representation of the e ects caused by traversing latent variables with SD-VAE and other state-of-the-art methods. 80

4.5 Construction of the plots for the evaluation of the per-attribute disentanglement. 82
4.6 E ects of traversing each latent variable across di erent mesh attributes with SD-VAE and other state-of-the-art methods. . . 83
4.7 SD-VAE results on bodies. 83
4.8 Complete latent traversals of SD-VAE trained on faces. 84
4.9 Complete latent traversals with state-of-the-art methods. 85
4.10 Complete latent traversals of SD-VAE trained on bodies. . . . 86
4.11 Ablation study of SD-VAE 87
4.12 Comparison of generalisation capabilities. 88
4.13 Latent interpolation experiment with SD-VAE. 89
4.14 Per-variable latent interpolation experiment with SD-VAE. . . . 89
4.15 Direct manipulation with SD-VAE. 90

5.1 Unoperated patients a ected by craniofacial syndromes. 96
5.2 Manifold visualisation and craniofacial syndrome classi cation. . 103
5.3 Example of surgical region de nition. 105
5.4 Spectra of real subjects. 107
5.5 Spectra of real subjects according to their age-group. 108
5.6 Spectra of real subjects according to their sex. 109
5.7 Evaluation of the syndrome-preserving data augmentation. . . . 110
5.8 Latent evaluation of SD-VAE trained on craniofacial data. . . . 112
5.9 Attribute-speci c latent evaluation of SD-VAE trained on craniofacial data. 114
5.10 Embedding of the surgical procedures that could be performed on a Crouzon patient. 115
5.11 Latent distances of the surgical procedures that could be performed on a Crouzon patient. 116
5.12 Monobloc and FOAR latent interpolations for a Crouzon patient. 118
5.13 Bipartition and Le Fort II latent interpolations for an Apert patient. 119

5.14 FOAR latent interpolations for a Muenke patient 119
5.15 Global interpolation trajectories determined by procedure-speci c latent-interpolations for an Apert and a Muenke patient. 120
5.16 Analysis of an atypical Crouzon patient 121

6.1 Examples of shape generation and editing with Local Eigenprojection Disentangled VAE (LED-VAE). 126
6.2 Schematic representation of the local eigenprojection 128
6.3 Schematic representation of LED-VAE 131
6.4 Local eigenprojection distributions 136
6.5 Random samples generated by LED-models, SD-VAE, and other state-of-the-art methods. 139
6.6 Random samples generated by LED-VAE models trained on other datasets. 140
6.7 Random samples and latent traversals of LED models and state-of-the-art methods. 141
6.8 Latent traversals of all latent variables with LED models and state-of-the-art methods. 142
6.9 Complete latent traversals of LED-Models. 143
6.10 Complete latent traversals of state-of-the-art methods. 144
6.11 Complete latent traversals of LED-VAE trained on other datasets. 145
6.12 Results of LED-VAE on other datasets. 146
6.13 Ablation study of LED-VAE. 147
6.14 E ects of LED-VAE smoothness weight on random samples and latent traversals. 148
6.15 E ects of LED-VAE local eigenprojection weights on random samples and latent traversals. 148
6.16 E ects of traversing each latent variable of LED-VAE trained enforcing latent disentanglement with di erent attribute segmentations. 149
6.17 Complete latent interpolations with LED-VAE and SD-VAE. . . 151

6.18 Per-attribute latent replacements with LED-VAE and SD-VAE. 151
6.19 Additional per-attribute latent replacements with LED-VAE . . 152
6.20 Additional per-attribute latent replacements with LED-VAE trained on body shapes. 152
6.21 Comparison of direct manipulation between LED-VAE and SD-VAE. 153

List of Tables

4.1 Quantitative comparison between the Swap Disentangled Variational Autoencoder and other state-of-the-art methods. 80

6.1 Quantitative comparison between the Local Eigenprojection Disentangled models and other state-of-the-art methods. 138

Chapter 1

Introduction

1.1 Problem Statement and Motivation

Digital humans are computer representations of human beings that have become central elements in many applications. In the movie, video-game, and advertisement industries, digital humans provide new opportunities to create more immersive and impactful experiences. They can be easily manipulated to t a speci c artistic vision, but more importantly, in movies, they can also be used to replace deceased actors or stunt performers in dangerous shots [201]. In Augmented Reality (AR) and in Virtual Reality (VR), they allow for the creation of interactive and lifelike experiences that can be used for entertainment, communication, or training purposes [227, 209]. Currently, the Metaverse is the VR application that is expected to disrupt the future of the web as we know it by allowing users to work, meet, play, and socialise in 3D virtual worlds [147]. Digital humans {who could be either controlled by real people or by Arti cial Intelligence (AI) techniques{ are arguably the most important element of the Metaverse, as they are going to be the main inhabitants of these virtual worlds. In healthcare, digital humans often require internal organs and they can be used to train healthcare professionals in real-life simulations of medical procedures, to enhance patient consultations, to perform surgical planning and simulation, to improve surgical procedures with Computer Assisted Surgery (CAS), and even for diagnostic purposes [134, 166]. Digital

humans are essential also for ergonomics studies, in the clothing industry, the military, in social media, and in many other elds.

Digital humans are structured objects with multiple components that change according to the application scenario. For example, when photo-realism is required, they are usually characterised by their shape, material and shading properties, as well as hair and clothes. Although some of these components are not always required, the geometric shape of a digital human, or part of it, is always present. Recent research has focused on investigating implicit shape representations, where continuous functions are used to represent volumes and surfaces [222, 223, 18, 189]. Although they can represent watertight surfaces with arbitrary surface topology and resolution, they are not yet optimised for editing, simulation, and rendering [38], making them di cult to integrate in many application scenarios. Therefore, even implicit shape representations are often converted into explicit representations, such as meshes and point-clouds. These more traditional representations, and meshes in particular, are the most widely adopted to represent the geometric shape of digital humans.

Creating digital humans is extremely di cult, especially when realism is required. A common method to create realistic digital humans is to digitise real people with scanning systems and capturing techniques [53, 227]. This can be achieved with complex multi-camera setups leveraging controlled lighting conditions, 3D scanners, medical imaging devices like Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), or even with a single camera in-the-wild. However, the quality and type of information that can be captured varies according to the acquisition method. When capturing a real human is not an option because having a digital double is not the main goal, digital humans need to be fully synthesised. Currently, this generation procedure is either manually performed by highly skilled artists or it involves consumer-level avatar design tools characterised by sets of sliders that are used to change the avatar's shape. However, digitally sculpting just the geometric shape of the head of a character can easily require a highly skilled digital artist weeks

to months of work [72], thus making the manual procedure viable only for a handful of users and application scenarios. On the other hand, the many semi-automated avatar design tools developed to make the generation process simpler and faster, all inherit the intrinsic constraints of their underlying generative techniques.

The most common generative models for digital human meshes are currently based upon blendshapes [125, 158, 195], Principal Component Analysis (PCA) [21, 168, 115], Variational Autoencoders (VAEs) [174, 69, 12, 44], or Generative Adversarial Networks (GANs) [39, 66, 112, 2]. Some of these models, like those based on PCA or Autoencoder (AE) architectures, are also often used for shape analysis purposes [100, 107, 162]. Interestingly, when Deep Learning (DL) models such as AEs, VAEs, and GANs operate on meshes, Geometric Deep Learning (GDL) techniques need to be used instead of the traditional DL-operators developed to operate in Euclidean domains. In addition, the aforementioned generative models all share one particular issue: the creation of local features is di cult or even impossible. In fact, not only do generative coe cients (or latent variables) lack any semantic meaning, but they also create global changes in the output shape. Similarly, when models based on PCA, AEs, or VAEs are used for shape analysis, they produce latent representations encoding global shape properties that are di cult to interpret, thus making local shape analysis impossible. For this reason, this thesis focuses on the creation of GDL-based models for shape analysis and generation that leverage latent disentanglement, which is a desirable property to o er control over local shape attributes.

By de nition [16], with a disentangled latent representation, changes in one latent variable a ect only one factor of variation while being invariant to changes in other factors. More interpretable and structured latent representations of data that expose their semantic meaning have been widely researched in the arti cial intelligence community in relation to 2D images [82, 96, 105, 55, 50, 205, 176], but this is still an open problem especially

for generative models of 3D shapes [12]. Most research on latent disentanglement of generative models for the 3D shape of digital humans addresses the problem of disentangling the pose and expression of a subject from its identity [12, 11, 44, 2, 219, 123, 222], but none of these works is able to provide disentanglement over the latent variables controlling the local attributes characterising the identity. Note that in this thesis, the identity is described by the pose and expression invariant intrinsic geometric properties of a digital human shape. As the identity is independent from extrinsic properties such as pose and expression, subjects are represented with a standardised pose (i.e., the \T" pose) and a neutral expression.

1.2 Aim of the Thesis

The overall goal of this PhD thesis is to improve the analysis and generation of digital human's shapes by proposing new latent disentanglement techniques for GDL-based models. With a growing interest in the Metaverse, simplied creation processes for diverse digital humans will become increasingly important. These processes will bene t experienced artists across multiple industries and, more importantly, will democratise the character generation process by allowing users with no artistic skills to easily create their unique avatars. In addition, the improved disentanglement will open new avenues for shape analysis, which, in this thesis, is applied in the medical domain in relation to severe craniofacial deformities.

The research hypotheses that guide this PhD work can be summarised as:

- Hypothesis 1 ($H1$): The latent disentanglement of local shape attributes can improve the control over the generation of 3D human shapes.
- Hypothesis 2 ($H2$): The latent disentanglement of local shape attributes can improve the analysis of human shapes and provide medical insights.

The two hypotheses are proved in relation to di erent application scenarios involving digital humans.

1.3 Structure of the Thesis

This dissertation is organised as follow:

Chapter 2 provides an overview of the background information and related works useful to better understand the following chapters.

Chapter 3 inspired the formulation of H_1 and describes how a GDL-based generative model trained on complete 3D shapes can be embedded in a shape completion procedure as an alternative to intraoperative registration. The algorithm is applied to laparoscopic liver Minimally Invasive Surgery (MIS). Given the preoperative shape of a liver and the partial point cloud representing its visible intraoperative surface, the proposed method predicts the intraoperative complete shape of the liver. Some level of disentanglement is imposed by construction as, during shape completion, latent shape properties are separately optimised from rotations and translations. However, the latent representation of the generative model is still entangled. In fact, traversing the latent variables highlighted their lack of semantic meaning as well as the lack of control over local shape attributes. These issues hindered the development of more advanced techniques for shape completion in liver surgery and motivated the research of new latent disentanglement techniques for GDL-based generative models.

Chapter 4 proves H_1 and details a novel method to obtain more disentangled, interpretable, and structured latent representations for 3D VAEs. The new disentanglement technique based on a curated mini-batching procedure and a new latent consistency loss was used to train the Swap Disentangled VAE (SD-VAE) on 3D shapes of heads and bodies.

Chapter 5 addresses H_2 by analysing head shapes of babies born with craniofacial syndromes. The SD-VAE introduced in Chapter 4 is also at the core of this work, which opens new avenues for craniofacial diagnosis and surgical planning.

Chapter 6 focuses again on H_1, proposing a new latent disentanglement method for generative models of head and body shapes. The newly introduced

Local Eigenprojection Disentangled Models (LED-Models) overcome some of the main limitations of SD-VAE and are an alternative approach particularly useful for shape generation.

Finally, chapter 7, compares the newly introduced methods, providing recommendations on their adoption with respect to potential applications, and illustrates future research directions.

Chapter 2

Background

This chapter provides the necessary technical background and context for the rest of the thesis. Not only does it present the current state of the art, but it also details the key techniques discussed throughout the thesis. As detailed in Sec. 1.2, the aim of the thesis is to improve the analysis and generation of digital human shapes by improving latent disentanglement in GDL-based data-driven models. Therefore, Sec. 2.1 de nes the shape representation used throughout this thesis, while Sec. 2.2 focuses on popular Statistical Shape Models (SSMs) for shape analysis and generation, Sec. 2.3 on GDL, Sec. 2.4 on latent disentanglement, and Sec. 2.5 on spectral geometry concepts widely adopted in the following chapters. Note that this thesis also encompasses solutions for medical applications. The description of medical-related background information is provided in the relevant chapters.

2.1 Shape Representation

In computer graphics and vision, three-dimensional shapes have multiple potential representations. Explicit representations such as voxelizations, point-clouds, and meshes are currently the most popular. Voxel-based representations (Fig. 2.1, left) are a discrete grid in the Euclidean space where each Cartesian coordinate identi es a unique point in the space, a voxel. To each voxel is assigned a value that de nes whether it is part of the object or of the background. In other words, voxels are three-dimensional pixels and vox-

Figure 2.1: Explicit shape representations. The same head is here represented with a voxelisation containing 3 million voxels in a 256^3 voxel grid, with a 2 thousand points point cloud, and with a 2 thousand vertices mesh.

elized objects are three-dimensional binary images. For this reason, voxel representations are very suitable to be processed by traditional neural network architectures whose operations, layers and metrics are the same of those used for images. Despite the many e orts to increase their resolution [71, 138, 204], voxelizations are still an ine cient volume representation because most of the important information should be captured on the surface and many voxels of the grid are then useless. For this reason, this thesis focuses on shapes represented as point clouds and especially as meshes.

Point clouds (Fig. 2.1,centre) are made of a set of three dimensional point coordinates that sample the surface of an object. With respect to voxelizations they are much more memory e cient, because points represent neither the inside nor the outside of the object, but only the boundary between these two regions: the surface. Thus, the reduced memory footprint eases computations, favours deeper neural network architectures, and results in higher quality shapes. Although this representation can o er increased resolution, the density/sparsity of the points becomes their main limitation.

Meshes (Fig. 2.1,right) are a discretization of the boundary surface of shapes represented by two-dimensional Riemannian manifolds embedded into the 3D Euclidean space $\mathbb{R}^3$) [26]. An n-dimensional Riemannian manifold can

be intuitively de ned as a smooth (i.e., di erentiable) multidimensional curved surface that is locally Euclidean, such that a small neighbourhood around any point on the manifold is di eomorphic to an open subset of an m-dimensional Euclidean space {where di eomorphic indicates that there is a continuously di erentiable mapping between the two spaces whose inverse is also continuously di erentiable [27]. To construct meshes, manifolds are discretized by sampling N points ($X \in \mathbb{R}^{N \times 3}$) that act as the vertices of the graph that gets constructed upon these points. The edges of this graph ($E \in \mathbb{N}^{N \times 2}$) represent the local connectivity of the manifold, telling whether two points belong to a neighbourhood or not. However, this discretization does not correctly capture the geometry of the underlying continuous manifold. A geometrically consistent discretization can be obtained by adding face information. Representing faces as triangles ($F \in \mathbb{N}^{N \times 3}$), manifold triangle meshes are obtained. In this thesis, the manifold triangle meshes used to represent 3D shapes and construct the GDL-based SSMs have a xed topology. By xing the topology, all meshes $M = \{X, E, F\}$ share the same edges and faces. Therefore, they di er from one another only by the position of their vertices X, which are assumed to be consistently aligned, scaled, and with point-wise correspondences with respect to a prede ned template mesh.

Recently, also neural implicit shape [140, 163, 139] and neural radiance elds [142, 145] representations have become increasingly popular. However, these representations are not compatible with most traditional computer graphics techniques (e.g., rendering, physical simulation, animation, etc.) and they are often further post-processed to be converted into an explicit shape representations.

2.2 Statistical Shape Models

Models for analysing and generating 3D shapes have been extensively researched in the computer graphics, computer vision, and medical imaging communities. When these models operate on human shapes (e.g., human organs, skeletal structures, heads, bodies, etc.) they are usually referred to as Statistical Shape Models (SSMs), which are models compactly describing collections of semantically similar objects in terms of variations from a mean shape [5]. The core idea of these models is to learn a mapping between 3D shapes and low-dimensional vectors. Once the mapping is established, shapes can be represented by small vectors and, vice versa, vectors can be transformed into 3D shapes. SSMs underpin anatomical cohort data analysis as they can provide compact and highly-discriminative encoding for a large set of shapes. This encoding produces feature vectors that are capable of capturing shape variations and that are well suited for the application of Machine Learning (ML) methods for classi cation, clustering, or for the identi cation of hidden structures and patterns in the shape data [5]. An appealing feature of SSMs is that they have generative powers. Not only they can create new shapes with weighted combinations of the shape instances that were used to create the model [5], but {under certain assumptions{ they can also become proper generative models. In fact, when the space of low-dimensional vectors representing the 3D shapes is associated to a probability distribution that measures the likelihood that a realistic 3D shape is represented by a particular vector, the SSM is also a generative model [53].

Blendshapes are a type of SSM that is still widely adopted for character animation or as consumer-level avatar design tool. By linearly interpolating between a prede ned set of artistically created shapes, the blend-weights can be easily interpreted [111]. However, to compensate for the limited exibility and diversity of these models, a large number of shapes are required. This makes the models very large and only a limited number of shapes can be used in most practical applications [72].

An alternative approach capable of o ering more exibility is to build models using PCA. These models are always built with 3D shapes that are registered (aligned) together and are in dense point correspondence. This allows the generation of meaningful and morphologically realistic shapes as linear combinations of training data. With the correct assumptions (see Sec. 2.2.1), they can also be regarded as generative models [196, 21, 53]. This technique was pioneered by [21] and further developed and adopted by many researchers [53]. Recently, [168, 167] combined multiple SSMs to create the rst combined, large-scale, full-head morphable model. In particular, the Universal Head Model (Uhm) [168] combines the Large-Scale Face Models (Lsfm) [22], which was built with face scans from 10,000 subjects, with the Lyhm head model [47]. In [167] it was extended by adding a detailed ear model, eye and eye region models, as well as basic models for mouth, teeth, tongue and the inner mouth cavity. PCA-based models and blendshapes are often combined. For instance, Smpl [125] learns linear PCA models of male and female body shapes from approximately 2,000 scans per gender, and subsequently uses the resulting principal components as body shape blendshapes capable of e ciently controlling the identity of a subject. The same approach is used also by Star [158], which not only creates more realistic pose deformations than [125], but it also leverages 10,000 additional scans to improve the generalisation capabilities of the model. Interestingly, methods like [151] and [59] developed PCA-like solutions relying on optimisation procedures and enforcing sparsity to achieve local control over head shape attributes. However, these SSMs are not generative models as they do not make any assumptions on the latent distribution.

The advent of geometric deep learning (Sec. 2.3) techniques brought a new set of operators making possible the creation of neural network architectures capable of processing 3D data such as point-clouds and meshes. [174] introduced the rst VAE for the generation of head meshes. In its comparison against PCA, the VAE model used signi cantly fewer parameters and exhibited

superior performance in generalisation, interpolation, and reconstruction. This pioneering work was followed by many other autoencoders which di ered from one another mostly by their application domain and the mesh operators used in their architecture [121, 61, 215, 225, 69, 48, 192, 25]. These mesh operators were also used for generative models based on GAN architectures [157, 39], but they appear to be less frequent than their VAE counterparts. Most GAN architectures operate in the image domain by representing 3D shapes in a UV space [144, 112]. GAN models are generally able to generate more detailed and realistic 3D shapes than autoencoders at the cost of being more unstable and di cult to train.

It is worth mentioning two additional types of models that recently become popular in the image domain and that are staring to be adopted also as SSMs: transformer networks and di usion models. For 3D shapes, multiple transformer architectures have been proposed [119, 118, 153, 91, 34]. Particularly relevant is the transformer-based AE operating on faces with arbitrary topologies and resolutions proposed in [34]. Despite being more computationally demanding, this architecture could be considered an alternative to the GDL-based models widely adopted throughout this thesis. On the contrary, di usion models {which can also leverage transformer architectures [153, 91]{ can not trivially replace the proposed models because the iterative denoising procedure makes them unsuitable for interactive shape generation and the high-dimensional latent space (retaining instead of reducing the input domain dimensionality) can not be used for shape analysis.

Before formally introducing the GDL operators making possible the creation of DL models for 3D shapes, the technical background necessary to understand how PCA, VAEs, and GANs operate is described in Sec. 2.2.1, Sec. 2.2.2, and Sec. 2.2.3 respectively.

2.2.1 Principal Component Analysis

Let $v_i \in \mathbb{R}^{3N}$ be the flattened vertex coordinates ($X_i \in \mathbb{R}^{N \times 3}$) of one of the M training shapes, and $V = [v_1^T; \ldots; v_M^T]^T \in \mathbb{R}^{M \times 3N}$ the matrix containing all the training data. In addition, let C be the data matrix obtained from V after centring each v_i by subtracting from each row of V the column-wise mean vector $\bar{v}$. C can then be standardised dividing the rows by the column-wise standard deviation vector σ. To compute the principal components, first, the covariance matrix $C^T C$ is eigendecomposed. Then, the d eigenvectors with the highest eigenvalues are selected as the principal components. Alternatively, the principal components can also be obtained by computing the Singular Value Decomposition (SVD) of C because its right singular vectors are equivalent to the eigenvectors of $C^T C$ and the singular values are equal to the square-root of the eigenvalues. Regardless of the technique used to compute them, the principal components are the vectors explaining most of the variance in the data and they can be used to compress new data or even to efficiently create new shapes as linear combinations of training shapes. In fact, the flattened vertex coordinates of a new shape can be obtained as:

$$v_{new} = \bar{v} + Ew$$

where $E \in \mathbb{R}^{3N \times d}$ is the matrix containing all the principal components and $w \in \mathbb{R}^d$ are the shape parameters (also known as latent variables in DL models). If C was standardised, Ew also needs to be multiplied by σ. Interestingly, the shape parameters (w) of head shapes, follow a multivariate normal distribution, which can be directly deduced from the eigenvalues corresponding to E [21, 53]. This enables the use of PCA as a generative model.

Note that PCA-based models are global models. Therefore, to offer local control, a separate model for each region needs to be created. As it can be observed in Fig. 2.2, the main issue of naive per-part methods, such as the bundle of PCA models, is that shape attributes are independently generated. Even though this makes the different attributes fully disentangled between each oth-

ers, signi cant surface discontinuities appear during the generation procedure. To overcome this issue, [21] blended the di erent regions at their borders and, more recently, [72] leveraged a constrained optimisation to generate new faces and interactively sculpt them.

Figure 2.2: Random samples generated by a bundle of PCA models trained each on a di erent shape attribute.

2.2.2 Variational Autoencoders

Variational Autoencoders are built as probabilistic encoder-decoder pairs parameterised by two separate neural networks (Fig. 2.3). The probabilistic encoder is de ned as a variational distribution $q(z|X)$ that approximates the intractable model posterior. It predicts the mean and standard deviation of a Gaussian distribution over the possible z values from which X could have been generated. The probabilistic decoder $p(X|z)$ describes the distribution of the decoded variable given the encoded one. Similarly to the encoder, it predicts a mean and a standard deviation of a Gaussian distribution. However, in practice, an identity covariance is assumed and the output of the decoder is just the mean, which can be regarded as the actual predicted data-point. During the generation process, a latent vector z is sampled from a Gaussian prior distribution $p(z) \quad N(0; I)$ and an output shape is generated by the probabilistic decoder. Since the decoder is used as a generative model, it is also referred to as generator. Following this convention, the VAE architecture is de ned as a pair of non-linear functions $\{E; G\}$, where $E : X \to Z$ maps from the vertex embedding domain X to the latent distribution domain Z, and $G : Z \to X$ vice versa (Fig. 2.3). and are the learnable parameters of these probabilistic functions.

Figure 2.3: VAE architecture. Given an input shape X, the probabilistic encoder E predicts and , which parameterise the variational distribution $q(z|X)$. A sample from this distribution, z, is then fed to the generator G, which e ectively tries to reconstruct the input shape. During training, a reconstruction loss encourages X^0 to match X and a Kullback{Leibler (KL) divergence pushes $q(z|X)$ towards $p(z)$. After training, new shapes can be generated sampling z from $p(z)$ and feeding it to the generator G.

E and G are jointly trained by maximising the variational lower bound, which is also known as Evidence Lower Bound (ELBO). Intuitively, the training objective for VAEs is to obtain a variational distribution $q(z|X)$ capable of properly approximating the intractable model posterior $p(z|X)$. The similarity of these two distributions is assessed using the KL divergence, which measures the expectation on the information difference between the distributions (see Technical Note 2.1) and it can be written as:

$$D_{KL}\left(q(z|X)\,||\,p(z|X)\right) = \int q(z|X)\log\frac{p(z|X)}{q(z|X)}\,dz. \tag{2.1}$$

Knowing that the KL divergence is always non-negative (i.e. $D_{KL}(\,;\,) \geq 0$), it is possible to derive the following:

$$\log p(X) \geq E_{z\sim q(z|X)}\log p(X|z) - D_{KL}\left(q(z|X)\,||\,p(z)\right) \tag{2.2}$$

The right hand side of this equation is the ELBO and, when maximised, it also maximises the log probability of the data. $E_{z\sim q(z|X)}\log p(X|z)$ is an expectation term that is approximated with Monte Carlo Sampling (see Technical Note 2.2). This term is also a reconstruction term measuring the likelihood of the data reconstructed by the generator G, while the KL divergence in the ELBO loss acts as a regulariser pushing the variational distribution towards the prior latent distribution. When the latent prior and variational distribution are approximated by Gaussian distributions, this KL divergence can be explicitly formulated as $L_{KL} = D_{KL}\left(q(z|X)\,||\,p(z)\right) = \frac{1}{2}\left(\sigma^2 + \mu^2 - \log(\sigma^2) - 1\right)$. The full derivation of the ELBO loss is available in [154].

Note that during training z needs to be sampled from the variational distribution, but directly sampling it as $z \sim q(z|X) \sim N(\mu\,;\,\sigma)$ is not possible as the sampling operation is not differentiable and gradients could not be backpropagated through E. Therefore, a reparametrisation trick needs to be used to differentiably sample z from the variational distribution. Being $\epsilon \sim N(0;I)$ a random vector drawn from a Gaussian distribution, then the

sampling operation can be implemented as $z = \quad + \quad$, where $[\ ;\ ] = E(X)$.

Also, to balance the contributions of the two terms in Eq. 2.2 a weighting constant is often added to the KL-divergence. Since manually selecting this constant can be di cult, recent methods de ned iteration dependants weights [10] or learned how to weight the two terms during training [17, 46].

Technical Note 2.1. Kullback{Leibler Divergence

As previously mentioned, the KL-divergence measures the expectation on the information di erence between two distributions. In information theory, the level of information of a random variable z distributed according to $p(z)$ can be measured as $I_p(z) = \quad \log p(z)$. As it can be observed from this formula, the higher the probability of an event, the lower its information content is. As highlighted in [154], this can be intuitively explained because if someone states something obvious (i.e., highly probable), than the statement is not informative.

Therefore, given two distributions $p(z)$ and $q(z)$, the expectation on their information di erence can be written as:

$$D_{KL}\ q(z)jj\,p(z)\ = E_{z\ \ q(z)}\ \ I\ = \int\ \ I_p(z)\ \ \ I_q(z)\ \ q(z)dz$$
$$= \int\ q(z)\ \log\frac{q(z)}{p(z)}dz:$$

Thanks to the logarithm properties, it can also be written like in Eq. 2.1 as:

$$D_{KL}\ q(z)jj\,p(z)\ = \int\ q(z)\ \log\frac{p(z)}{q(z)}dz: \tag{2.3}$$

Interestingly, the KL-divergence is also called relative entropy, as it can be formulated as $D_{KL}\ q(z)jj\,p(z)\ = H\ \ q(z);p(z)\ \ \ H\ \ q(z)$, with $H\ \ q(z)\ = \int \log q(z)dz$ being the entropy of $q(z)$, and $H\ \ q(z);p(z)\ = \int q(z)\ \log p(z)dz$ being the cross-entropy of the two distributions. This can be seen as the amount of information lost when the the distribution $p(z)$ approximates $q(z)$ [31].

Technical Note 2.2. Monte Carlo Sampling

Monte Carlo methods can be used to solve stochastic as well as deterministic, but intractable, problems. Interestingly, this technique was developed in statistical physics during the development of the atomic bomb [146]. In this thesis, it is mostly used to compute expectations within loss functions. Like reported in [109], when a sum or an integral s cannot be computed exactly, they are usually approximated with Monte Carlo sampling. Being

$$\begin{aligned} s = E_p[f(x)] &= \sum_x p(x) f(x) \quad \text{(if } p(x) \text{ is a probability distribution)} \\ &= \int_x p(x) f(x) dx \quad \text{(if } p(x) \text{ is a probability density)} \end{aligned}$$

its value can be approximated by drawing n samples from p and then computing the empirical average

$$\hat{s}_n = \frac{1}{n} \sum_{i=1}^{n} f(x^{(i)}):$$

Following the law of large numbers, if the samples are independent and identically distributed, and the variance of the individual terms $Var[f(x^{(i)})]$ is bounded, then $\hat{s}_n$ converges to s for n going to in nity. The uncertainty in the Monte Carlo approximation can be computed as:

$$Var[\hat{s}_n] = \frac{1}{n^2} \sum_{i=1}^{n} Var[f(x^{(i)})] = \frac{Var[f(x)]}{n}:$$

2.2.3 Generative Adversarial Networks

To date, GANs (Fig. 2.4) are one of the most popular generative models as they demonstrated incredible results especially in the image domain [94, 95, 93, 92]. Although many di erent avours have been proposed [135, 7, 73, 94, 225], they are all built following the same underlying principle rstly introduced in [70]. In fact, GANs are built as a pair of neural networks trained while competing

Figure 2.4: GAN architecture. Given a random vector z sampled from a Gaussian distribution $p(z)$, the generator G, generates a sample X^0. While the discriminator D learns to distinguish real data X from generated samples X^0, the generator learns to create progressively better shapes. Note that the shape here used to represent X^0 was created for the purpose of illustrating the concept of an unrealistic human head shape. In early stages of training, output meshes are noisy and with self-intersections, while in late stages it should be impossible to distinguish real from generated samples.

against each other in a zero-sum game. The generator network G takes a latent vector z, usually sampled from an isotropic unit Gaussian, and generates a new sample as $X^0 = G(z)$. Its adversary, usually called discriminator D, attempts to distinguish the generated samples from the training data. During training, each player attempts to maximise its own payo function: $v(G; D)$ for the discriminator, and $v(G; D)$ for the generator. Therefore, the training objective can be written as:

$$\arg\min_G \max_D v(G; D):$$

With the default payo choice, it becomes:

$$\arg\min_G \max_D E_{X \quad p_{data}} \log D(X) + E_{z \quad p(z)} \log 1 \quad D(G(z)) \quad ; \qquad (2.4)$$

which is e ectively a binary cross-entropy loss based on the assumption that, during the discrimination, real data are labelled with a value of 1, and generated samples with a value of 0. Note that, while the discriminator has control over both terms, the generator has control over the second term only.

With the objective in Eq. 2.4, while the discriminator tries to learn how to distinguish real from generated samples by maximising the entire equation, the generator attempts to fool the discriminator by learning to minimise $E_{z \sim p(z)} \log 1 - D(G(z))$. When convergence is reached, generated samples are indistinguishable from real ones and the discriminator outputs $\frac{1}{2}$ everywhere as it is incapable of distinguishing what is real and what is not.

Even though GANs are intuitive and apparently easy to train, in practice convergence is not guaranteed as they are unstable and su er from vanishing gradients and mode collapse (i.e., the generator generates only one or a small set of samples). Due to the training di culties, this thesis focuses mostly on methods for VAE architectures. Nevertheless, Ch. 6 introduces a latent disentanglement technique that is also suitable for GAN-based SSMs.

2.3 Geometric Deep Learning

As mentioned in Sec. 2.1, the triplet made of vertex positions, edges and faces $(X; E; F)$ is referred to as triangular mesh and is a discretisation of a Riemannian manifold embedded in $\mathbb{R}^3$. The non-Euclidean nature of such data implies that there are no familiar properties such as global parameterization, a common system of coordinates, a vector space structure, or shift invariance. For this reason, basic operations that are taken for granted in the Euclidean case (e.g. convolution, pooling and unpooling) are not well de ned on non-Euclidean domains [28].

Geometric Deep Learning is an umbrella term for the emerging research eld that aims at generalising DL to non-Euclidean domains such as graphs, manifolds, point clouds and meshes. Convolutions suited for these domains are currently the most researched operators in the eld. There are two possible approaches: leverage the convolution theorem and perform convolutions in the spectral domain, or leverage spatial properties of the 3D shapes to perform convolutions in the spatial domain. The rst formulation of spectral convolution [29] was very computationally expensive and it was also explic-

itly leveraging the graph Laplacian operator (Sec. 2.5). Therefore, the Fourier basis computed from the Laplacian were domain dependant and convolutions could not be generalised across di erent topologies. To reduce the risk of over tting, the number of learnable parameters was reduced by restricting the amount of spectral multipliers to those corresponding to localised lters. Instead of explicitly operating in the spectral domain with spectral multipliers, [49] proposed to use a Chebyshev polynomial expansion to e ciently represent localised lters and [99] proposed a more simpli ed version of it. Both lters can be interpreted as operating in the spatial domain and are capable of dealing with di erent topologies. Convolutions explicitly created to operate in the spatial domain have to deal with the lack of shift-invariance (each patch operator would depend on its position), but are able to generalise to di erent topologies. Geodesic methods take advantage of the manifold property of being locally Euclidean and de ne convolutional kernels on local polar systems of coordinates in the tangent space [137]. Anisotropic methods on the other hand still take advantage of local chartings, but they use lters similar to isotropic heat kernels which have the same e ects in all directions. A hybrid approach was also proposed in [143], where Gaussian kernels were learned on local pseudo coordinate systems de ned for each point. Alternatively, rather than relying on prede ned static coordinates, [200] proposed to dynamically compute correspondences from features learned by the network. Another popular approach to perform convolutions on 3D data is to leverage PointNet [171, 172] and its variants. These operators treat points independently and subsequently aggregate features by leveraging symmetric functions to obtain permutation invariance. However, these point-cloud-based convolutions cannot capture local features because they neglect the geometric relationships among points [207]. This drawback can be overcome by dynamically aggregating edge features of neighbouring points [207] or by approximating geodesic coordinates in the tangent space and representing the surface as an oriented point-cloud [191]. Finally, a recent approach was proposed for problems with a xed topology. Despite

this considerable limitation, [69] managed to achieve state of the art results with computationally e cient and very intuitive operators arbitrarily de ned on the surface of meshes.

Before providing technical details on the operators most used in this thesis (Sec. 2.3.1 Sec. 2.3.2 and Sec. 2.3.3), it is worth mentioning that GDL allowed the creation of better performing and non-linear generative models [174]. In addition, it revolutionised shape analysis. In fact, the extraction of shape descriptor features was previously a necessary pre-processing step to train other ML models. Now, end-to-end GDL-models can be directly used to analyse shape data [78].

2.3.1 Feature Steered Graph Convolutions

The Feature-steered graph convolutions de ned in [200] dynamically assign lter weights to k-ring neighbourhoods according to the features learned by the network. In particular, given a generic feature vector eld where each vertex i has a feature vector x_i, it is possible to de ne the output of the convolutional operator as

$$y_i = b + \frac{1}{|N_i|} \sum_{j \in N_i} \sum_{m=1}^{M} q_m(x_i; x_j) W_m x_j \tag{2.5}$$

where b is a learnable bias, $q_m(x_i; x_j)$ is a translation-invariant assignment operator that, using a soft-max over a linear transformation of the local feature vectors, learns to dynamically assign x_j to the m-th learnable weight matrix W_m, and N_i is the neighbour of the i-th vertex with cardinality $|N_i|$.

2.3.2 Spiral++ Convolutions

In this thesis, the most used convolutional operators are the Spiral++ convolutions introduced in [69]. Although this is an arbitrary choice and other operators could be used, Spiral++ convolutions are intuitive and particularly well suited for SSMs as they are designed to e ciently operate on datasets of meshes in dense point correspondence. The creation of spiral sequences is at

the core of this operator. Spirals are a simple yet e ective approach to aggregate neighbouring mesh vertices into ordered sequences. Given a vertex, the spiral sequence is obtained by arbitrarily selecting one neighbour and following a clockwise spiral until the spiral length is reached. The receptive eld of these convolutional operators can be expanded by dilating the spirals (i.e. not selecting certain vertices along the sequence). Denoting by $S(n; l)$ the spiral centred at vertex n with length l, the convolution at layer k is de ned as:

$$x_n^{(k)} = \mathrm{MLP}^{(k)} \underset{j \in S(n;l)}{\mathbin{\|}} x_j^{(k-1)}$$

where $\|$ is the concatenation operation over the vertices in the spiral $S(n; l)$, $x_n^{(k)}$ are the vertex features at layer k, and MLP is a multilayer perceptron. Note that spirals are xed during training because they are pre-computed only once for all vertices.

2.3.3 Pooling and Un-pooling with Fixed Topologies

The pooling operators used in this thesis were rst introduced in [174]. The operators are precomputed during a quadric sampling procedure [65] applied to the mean shape of the training data $\mathcal{M}$ that iteratively contracts the vertex pair with the smallest quadric error. During this procedure both a pooling and an un-pooling sparse matrix are de ned. In particular, the pooling matrix $Q_d \in \{0; 1\}^{N_{k+1} \times N_k}$ is a sparse matrix where $Q_d(p; q) = 1$ if vertex q has been preserved during quadric sampling and $Q_d(p; q) = 0$ otherwise. The un-pooling matrix $Q_u \in \mathbb{R}^{N_k \times N_{k+1}}$ leaves the preserved vertices unchanged by setting $Q_u(q; p) = 1$. Contracted vertices are expressed in barycentric coordinates with respect to the closest preserved triangle, and then their corresponding elements in Q_u are set to the barycentric weights. This makes possible to restore the contracted vertices. These precomputed sparse matrices are matrix multiplied with the vertex features computed by the di erent network's layers to achieve pooling and un-pooling.

2.4 Latent Disentanglement

As mentioned in Sec. 1.1, with a disentangled latent representation, changes in one latent variable affect only one factor of variation while being invariant to changes in other factors. However, research on latent disentanglement often focuses on the scenario in which only raw observations are available without any supervision about the generative factors, and it is usually performed on images [106, 96, 105, 55, 50, 176, 205]. The β-VAE [82] is probably the simplest model used to improve disentanglement in a VAE [98]. By increasing the weight of the KL divergence, β-VAE showed better latent disentanglement properties at the expense of a reduced quality of the generated samples (see Sec 2.4.1). Subsequent work, such as [106, 96], tried to overcome this limitation. The Disentangled Inferred Prior VAE (DIP-VAE) [106] leverages an additional regularisation term on the expectation of the approximate posterior over observed data (see Sec. 2.4.2). The Factor VAE [96] encourages the latent distribution to be factorial, and therefore independent across dimensions, by using a latent discriminator and by adding a total correlation term in the VAE loss function (see Sec. 2.4.3). An interesting approach to encourage latent variables to represent predefined transformations was proposed in [105], where mini-batches are created combining active and inactive transformations and gradients influencing the latent are modified during backpropagation. However, this method requires synthetic datasets created with known properties that can be used during training to achieve the disentanglement. Recently, [55] proposed a VAE in which the objective function is hierarchically decomposed to control the relative levels of statistical independence between groups of variables and for individual variables in the same group. The recursive formulation of the loss introduces additional terms for any variable that has to be disentangled and works only where the factors of variation are uncorrelated scalar variables, a requirement that hampers the applicability of the model in real-world scenarios. Finally, the Guided-VAE [50] in its unsupervised setting leverages a secondary decoder that learns a set of PCA bases that are used

to guide the training over simple geometrical shapes. Nevertheless, being the secondary decoder based on a PCA, latent variables su er the same problems as PCA models.

In the 3D realm, there are currently two prominent streams of research: the one disentangling the identity from the pose or expression of digital humans [12, 11, 44, 219, 224, 193, 89, 86, 160], and the stream attempting to disentangle parts of man-made objects [214, 149, 114, 177]. In both cases, the proposed solutions require complex architectures. In addition, in the former category, current state-of-the-art methods do not attempt to disentangle local shape attributes. The latter category appears better suited for this purpose, but the type of generated shapes is substantially di erent because the generation of object parts needs to consider intrinsic hierarchical relationships, and surface discontinuities are not a problem.

Like for VAEs, the research on GANs comes mostly from the imaging domain, where good levels of control over the generation process were recently made possible. Most of these models leverage segmentation maps [87, 110, 120], additional attribute classi ers [79, 185], text prompts [173], or manipulate the latent codes and the parameter space of the pre-trained model to achieve the desired results [93, 75, 184, 120]. While the rst two approaches require multiple inputs and additional supervision, the last two o er limited editing exibility. In fact, describing the shape of human parts is a di cult task that would ultimately limit the diversity of the generated shapes, while the post-training manipulation may limit the exploration of some latent regions. Only a few methods explicitly seek disentanglement during training [4, 202]. However, [4] is speci cally designed for grid-structured data, like images, and [202] still requires a pre-trained GAN and two additional networks for disentanglement. In the 3D shapes domain, GAN disentanglement is still researched to control subject poses and expressions [37, 157] or object parts [113]. However, they su er the same problems described for 3D VAEs: they have complex architectures and do not have control over the generation of local shape attributes

characterising the identity of digital humans.

An alternative, yet related, stream of research is focused on the creation of capsule neural networks. These networks were born in the computer vision community with an inverse graphics objective, where images should be automatically deconstructed into parts and their relationships. From their introduction in the transforming autoencoders [83], multiple alternative approaches were proposed, where the introduction of dynamic routing [182] and of Matrix capsules with EM routing [84] are arguably the most explored alternatives. While most capsule networks operate in the 2D domain, some 3D alternatives were also introduced [220, 221, 210, 190]. These methods aim at nding local relationships between shapes and latent representations in order to improve the local analysis and generation of shapes. However, capsules require sub-networks which separately process di erent parts. While several disentangling methods for 3D shapes requiring sub-architectures may be associated to capsule networks, the methods proposed in this thesis are capable of achieving the same goal without requiring independent sub-networks. Sec. 2.4.1, Sec. 2.4.2, and Sec. 2.4.3 introduce the most popular avours of VAE with a disentangled latent representation: the -VAE, DIP-VAE, and Factor VAE.

2.4.1 -VAE

In short, a -VAE [82] is a VAE trained with an increased weighting on the KL-divergence between variational and prior distribution. In fact, the ELBO loss that is maximised during training, and previously introduced in Eq. 2.2, can be rewritten as:

$$E_{z \quad q\,(z|X)} \log p\,(X|z) \qquad D_{KL} \quad q\,(z|X)||p(z) \;: \tag{2.6}$$

In [82], this equation is not directly derived from the KL-divergence between variational and posterior distribution (Sec. 2.2.2). Instead, their initial assumption is that in order to train an unsupervised generative model capable of generating new data, a suitable objective is to maximise the marginal like-

lihood of the observed data in expectation over the whole latent distribution, which can be formally written as:

$$\max E_{z \quad p(z)} \; p \; (X \, j z) \; :$$

Being $q \; (z j X \,)$ the inferred posterior con guration of latent factors given an observation X, they wish to obtain an inferred posterior capable of disentangling factors of variation. This is achieved by introducing a constraint on the inferred posterior, which is encouraged to match a prior distribution with independent variables. The selected prior distribution is an isotropic unit Gaussian distribution $p(z) = \; N \; (0; I \,)$, where the di erent dimensions are independent by construction {as a consequence of having unitary variance and zero covariance. Considering that during training z is sampled from the inferred posterior, the objective function can be re-written as:

$$\max_{;} E_{X \quad data} \left[E_{z \quad q \; (z j X \,)} \; p \; (X \, j z) \right] \qquad \text{subject to: } D_{KL} \quad q \; (z j X \,) j j \, p(z) \quad < \; :$$

Using the Lagrange multipliers, this can be easily formulated as in Eq. 2.6. Note that when $= 1$ the derivation of the $-$VAE can be considered an alternative derivation of the ELBO loss introduced in Sec. 2.2.2. However, $-$VAEs are required to have 1. This is because, when the training data contain independent factors of variation, 1 encourages the model to automatically learn a disentangled latent representation without any supervision.

Although 1 is the most common condition reported in literature to derive a $-$VAE, it is worth considering that the magnitude of this weight is actually in uenced by the assumptions made in the data term. The standard assumption, that enables the derivation of a $L2$ loss from the log probability of the data (Eq. 2.2 and Eq. 2.6), is that the noise model of the data term has a unitary covariance. If a di erent covariance is used, the weight needs to be changed accordingly.

2.4.2 Disentangled Inferred Prior VAE

As the name suggests, the DIP-VAEs [106] are VAEs achieving disentanglement through the inferred prior distribution. Following the law of total probability (which is the one commonly used to explicitly compute the denominator of the Bayes' rule), the inferred prior (or expected variational posterior) is de ned as:

$$q(z) = \int q(z|X)p(X)dX;$$

where $p(X)$ is the true data distribution, which is usually unknown and from which some samples (training data) are usually available. To achieve disentanglement, the inferred prior should be factorizable along its dimensions: $q(z) = \prod_i q_i(z_i)$. As mentioned in Sec. 2.4.1, an isotropic unit Gaussian satis es this requirement by construction. On simple datasets, a VAE with an isotropic unit Gaussian prior $p(z) = \mathcal{N}(0; I)$ can thus achieve some level of disentanglement, but on more complex datasets more supervision is required. Supervision can be provided by minimising a distance between the prior $p(z)$ and the inferred prior $q(z)$. Therefore, instead of maximising the ELBO loss as in Eq. 2.2, the objective is modi ed to consider also this additional term on the inferred prior:

$$\mathbb{E}_{z \sim q(z|X)} \log p(X|z) \quad D_{KL}\left(q(z|X) \| p(z)\right) \quad D\left(q(z) \| p(z)\right)$$

The obvious metric to use to evaluate $D\left(q(z) \| p(z)\right)$ would be the KL divergence introduced in Technical Note 2.1. Unfortunately, in this case the KL divergence has no closed form expression and another criteria needs to be used. The criteria proposed by [106] is based on matching the covariance of the two distributions. The covariance of the inferred prior is de ned as:

$$\mathrm{Cov}_{q(z)}[z] := \mathbb{E}_{q(z)}\left(z \quad \mathbb{E}_{q(z)}[z]\right)\left(z \quad \mathbb{E}_{q(z)}[z]\right)^T :$$

By the law of total covariance, and considering that the variational distribution represented by the probabilistic encoder E of the VAE predicts $[\ ;\ ] = E(X)$, the covariance of the inferred prior can be rewritten as:

$$\mathrm{Cov}_{q(z)}[z] = E_{p(X)}\ \mathrm{Cov}_{q(z|X)}[z] + \mathrm{Cov}_{p(X)}\ E_{q(z|X)}[z] = E_{p(X)}[\ ] + \mathrm{Cov}_{p(X)}[\ ]: \tag{2.7}$$

To match this covariance to the one of the prior, which is an isotropic unit Gaussian, it is su cient to compute the l_2-norm between $\mathrm{Cov}_{q(z)}[z]$ and the identity matrix. As is usually a diagonal matrix, the cross-correlations between the latents are due to only $\mathrm{Cov}_{p(X)}[\ ]$. This leads to two alternative formulations: the DIP-VAE-I, which considers only $\mathrm{Cov}_{p(X)}[\ ]$, and the DIP-VAE-II, which considers the full $\mathrm{Cov}_{q(z)}[z]$ in Eq. 2.7. Therefore, the metrics used to evaluate the similarity between prior and inferred prior can be written as:

$$D\ q(z)||p(z) = {}_{od} \sum_{i \neq j} \mathrm{Cov}_{p(X)}[\ ]_{ij}^{2} + {}_{d} \sum_{i} \mathrm{Cov}_{p(X)}[\ ]_{ii} \quad 1\ ^{2}; \tag{2.8}$$

$$D\ q(z)||p(z) = {}_{od} \sum_{i \neq j} \mathrm{Cov}_{q(z)}[z]_{ij}^{2} + {}_{d} \sum_{i} \mathrm{Cov}_{q(z)}[z]_{ii} \quad 1\ ^{2}; \tag{2.9}$$

where Eq. 2.8 is the metric used for DIP-VAE-I and Eq. 2.9 the one used for DIP-VAE-II. Both metrics separately evaluate the distance of the elements on the diagonal and o the diagonal, which are weighted with ${}_{d}$ and ${}_{od}$ respectively.

2.4.3 Factor VAE

The Factor VAE is another variant of the traditional VAE previously described in Sec. 2.2.2. This model achieves latent disentanglement by encouraging the distribution of latent representations to be factorial and hence independent across dimensions [96]. Similarly to the DIP-VAEs, disentanglement is achieved with the introduction of an additional term in the ELBO loss func-

tion, which can be rewritten as:

$$E_{z \sim q(z|X)} \log p(X|z) \quad D_{KL}\left(q(z|X) \,\|\, p(z)\right) \quad D_{KL}\left(q(z) \,\|\, q(z)\right); \qquad (2.10)$$

where is a weighting term, $q(z) = E_{p_{data}(X)}[q(z|X)] = \frac{1}{N}\sum_{n=1}^{N} q(z|X^{(n)})$ is the distribution of the latent representations for the entire training set, and $q(z) := \prod_i q(z_i)$ is a factorial distribution. The additional KL divergence in Eq. 2.10 (rightmost term) is called Total Correlation (TC) and, unfortunately, it is intractable because of the large number of components in the two distributions. Based on the observation that it is possible to sample both distributions, their KL divergence can be computed using the density ratio trick, which uses a discriminator D to approximate the density ratio arising in the KL divergence. Knowing that $q(z)$ can be sampled by choosing a data-point $X^{(n)}$ uniformly at random and then sampling $q(z|X)$, while $q(z)$ by sampling a mini-batch from $q(z)$ and randomly permuting across the batch for each latent dimension , it is possible to formulate the TC as:

$$TC(z) = D_{KL}\left(q(z) \,\|\, q(z)\right) = E_{q(z)}\left[\log \frac{q(z)}{q(z)}\right] \quad E_{q(z)}\left[\log \frac{D(z)}{1 - D(z^0)}\right]; \qquad (2.11)$$

where z^0 are the vectors in the mini-batches of permuted latents, thus sampled from $q(z)$.

The discriminator is trained to classify between samples of $q(z)$ and $q(z)$. VAE and discriminator are trained simultaneously by maximising Eq. 2.10 with the TC approximation (Eq. 2.11).

2.5 Spectral Geometry

Spectral mesh processing has played an essential role in shape indexing, sequencing, segmentation, parametrisation, correspondence, and compression [218]. Spectral methods usually leverage the properties of the eigenstructures of operators such as the mesh Laplacian. Even though there is no unique de nition for this linear operator, it can be classi ed either as ge-

ometric or combinatorial. Geometric Laplacians are a discretisation of the continuous Laplace-Beltrami operator [36] and, as their name suggests, they encode geometric information. Their eigenvalues are robust to changes in mesh connectivity and are often used as shape descriptors[175, 64]. Since they are isometry-invariant, they are used also in VAEs for identity and pose disentanglement [12, 11]. However, being geometry dependant, the Laplace-Beltrami operator and its eigendecomposition have to be precomputed for every mesh in the dataset. On the other hand, combinatorial Laplacians treat a mesh as a graph and are entirely de ned by the mesh topology. For these operators, the eigenvectors can be considered as Fourier bases and the eigenprojections are equivalent to a Fourier transformation [186] whose result is often used as a shape descriptor. If all shapes in a dataset share the same topology, the combinatorial Laplacian and its eigendecomposition need to be computed only once. For this reason, multiple graph and mesh convolutions [29, 49] as well as some data augmentation techniques [61] and smoothing losses [62] are based on combinatorial Laplacian formulations.

Both categories of Laplacian are built following the same underlying principle. In fact, a mesh Laplacian operator L locally takes the di erence between the value of a function f at a vertex and a weighted average of the function's values at the vertices of the one-ring neighbours. This can be formally written as

$$(Lf)_n = b_n^{1} \sum_{m \in N_n} w_{mn}(f_m \quad f_n);$$

where w_{mn} are symmetric edge weights, and b_n is a constant. Note that this is a local formulation referred to the n-th verthex, the global formulation can be expressed using the following matrix notation:

$$Lf = B^{1}Wf; \tag{2.12}$$

where B is a diagonal matrix whose entries are the b_n constants, and W is a symmetric matrix with diagonal entries equal to $\sum_{m \in N_n} w_{mn}$ and o diagonal

entries equal to w_{mn}. It can be shown that the eigenvectors of L are mutually orthogonal [218], a desirable property in many applications.

Since SSMs are built with datasets of shapes sharing the same topology, this thesis relies on combinatorial Laplacians. In particular two formulations are used: the Kircho and the Tutte Laplacians. The Kircho Laplacian is also commonly called graph Laplacian and it is built as the di erence between the degree matrix $D \; 2 \; N^N \; {}^N$ and the adjacency matrix $A \; 2 \; N^N \; {}^N$. Using the notation previously introduced in Sec. 2.1, where E are the edges of a mesh with N vertices, the adjacency matrix is de ned as

$$A_{mn} = \begin{cases} 1 & \text{if } (m;n) \; 2 \; E; \\ 0 & \text{otherwise,} \end{cases}$$

and the degree matrix D as a diagonal matrix where $D_{mm} = \sum_m A_{mn} = jN_m j$. Therefore, the Kircho Laplacian can be formally computed as

$$K = D \quad A: \tag{2.13}$$

Note that the formulation in Eq. 2.13 is equivalent to the one in Eq. 2.12 when the diagonal elements of B are $b_n = 1$ and $W = D \quad A = K$.

The Tutte Laplacian, which is the other operator frequently used in this thesis, can be derived as:

$$T = D \;{}^1K = D \;{}^1(D \quad A) = I \quad D \;{}^1A: \tag{2.14}$$

In this case, the the relation between Eq. 2.14 and Eq. 2.12 is more apparent as $B = D$ and $W = K$.

2.6 An Overall Perspective on the Topics

This chapter has provided an overview of the background information and relevant literature that are necessary to fully understand the methods detailed in the following chapters. Among the di erent shape representations introduced in Sec. 2.1, meshes are the preferred ones to train any of the models proposed throughout this thesis. The most common SSMs are undoubtedly VAEs (Sec. 2.2.2), but in Chapter 6 two di erent avours of GANs (Sec. 2.2.3) are also proposed. All the architectures leverage the GDL operators described in Sec. 2.3. In particular, the same pooling and unpooling operators presented in Sec. 2.3.3 are always used, while the feature steered graph convolution (Sec. 2.3.1) is used in Ch. 3 and the spiral++ convolution in Ch. 4, Ch. 5, and Ch. 6. The latent disentanglement methods presented in Sec. 2.4 are mostly used to benchmark the disentanglement performances of the methods proposed in Ch. 4 and Ch. 6, while the spectral geometry operators introduced in Ch. 2.5 are leveraged either to de ne smoothness losses or to propose new data augmentation techniques across the di erent chapters.

Chapter 3

Intraoperative Shape Completion

This chapter describes how a GDL-based generative model can predict the complete surface of a liver, given a partial point cloud of the organ obtained during a surgical laparoscopic procedure. The core of the proposed method is a VAE trained to learn a latent space for complete shapes of the liver. At inference time, the generative part of the model acts as a prior embedded in an optimisation procedure where the latent representation is iteratively updated to generate a shape that matches the intraoperative partial point cloud. The e ect of this optimisation is a progressive non-rigid deformation of the initially generated shape. In this case, some level of disentanglement is imposed by construction as, during the optimisation, latent shape properties are separately optimised from rotations and translations. Also, to compensate for the limited size of the liver dataset, a new data augmentation technique that randomly perturbs shapes in their frequency domain is introduced.

A paper on this work has been published as [61]: Simone Foti, Bongjin Koo, Thomas Dowrick, Jo-ao Ramalhinho, Moustafa Allam, Brian Davidson, Danail Stoyanov, Matthew J. Clarkson. Intraoperative Liver Surface Completion with Graph Convolutional VAE . Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis, pp. 198-207. Springer, Cham, 2020The -

nal authenticated version is available online at https://doi.org/10.1007/978-3-030-60365-6_19.

This Chapter starts by providing background information on Computer Assisted and Minimally Invasive Surgery (Sec. 3.1.1) and subsequently introduces the proposed method in the context of computer assisted procedures of the liver (Sec. 3.1.2). Then, the method is formally described (Sec. 3.2), experiments and results are presented (Sec. 3.3), and discussed (Sec. 3.4).

3.1 Introduction

3.1.1 Computer Assisted and Minimally Invasive Surgery

Computer Assisted Surgery (CAS), also known as Computer Aided Surgery or Computer Assisted Intervention, encompasses computer science methods such as AR, ML, multimodal imaging, and robotics to augment clinicians and provide better clinical outcomes [188]. Until recently, these terms were often confused with Image Guided Surgery and could have di erent interpretations. Following [67], who provided an o cial de nition by seeking consensus among experts of the eld, CAS can be de ned as \the broad use of information technology frameworks to enhance physicians' skills and augment senses (e.g., image-guided surgery), cognition (e.g., deep learning, machine learning), and execution (e.g., mechatronic, imaging and surgical robotics) with the aim of providing more precise and safer procedures".

CAS made its appearance in 1895, eight days after Roentgen's rst paper on X-rays, when it was used to remove a needle from a patient's hand. In the past century it was mostly used for neurosurgery, because the skull can guarantee a rigid and stable frame of reference, but its basic principles can be applied to any part of the body. In fact, CAS has current applications in neurosurgery, orthopaedics, plastic surgery, cardiology, as well as abdominal, kidney, liver, and prostate interventions [42]. Nowadays, in most of these applications there is a drive towards replacing open surgical procedures with

Minimally Invasive Surgery (MIS).

MIS is performed through small incisions in order to reduce surgical trauma and morbidity. When MIS procedures are performed in the abdominal or pelvic cavities, the abdomen is insu ated with gas to create a working volume (pneumoperitoneum) into which an endoscopic camera (laparoscope) and the surgical instruments can be inserted [133]. Despite the many bene ts of laparoscopic procedures, they introduce an additional level of complexity for surgeons. Hence, CAS techniques help and guide surgeons during the intervention. Many CAS approaches involve the use of AR, a variation of VR that has been proposed as a method of guidance in which the user sees the real world with virtual objects superimposed upon or composited with the real world [13]. The introduction of AR in MIS alleviates the loss of direct vision and tactile feedback that arises in this type of interventions. Laparoscopic AR provides intraoperative guidance with a rapid identi cation of subsurface targets (e.g., tumours, infections, foreign bodies) and critical structures (e.g., vessels, nerves, other organs). Even though AR was successfully deployed in surgical procedures where organs and tissues can be approximated with rigid structures (e.g. in neurosurgery), augmenting the laparoscopic scene of abdominal surgeries remains an open challenge because of the non-rigidity of the tissues and organs involved. This problem is accentuated especially in MIS of the liver, one of the biggest and most deformable organs [19]. From 2003, a few surgical procedures deploying these technique have been reported in literature. In particular, [103] performed 20 laparoscopic interventions on spleen, stomach, gallbladder or thoracic cavity and [197, 187, 194, 148] independently performed urological procedures on multiple patients. In more recent years, a robotic AR-assisted liver surgery and a pre-clinical evaluation of AR laparoscopic liver resection have been reported in literature [165, 108]. To date, most AR laparoscopic procedures are still considered research works where the augmentation is still performed manually or semi-automatically. In addition, the few cases that tried to deal with tissues deformation used only local compen-

sations, thus limiting the accuracy of the augmentations and the validation of the methods [19].

3.1.2 Shape Completion for Computer Assisted Liver Surgery

As mentioned in Sec. 3.1.1, the loss of direct vision and tactile feedback in laparoscopic procedures introduces an additional level of complexity for surgeons. AR is a promising approach to alleviate these limitations and provide guidance during the procedure. However, it remains an open challenge for laparoscopic surgery of the liver, which is one of the largest and most deformable organs. AR is usually achieved by registering a preoperative 3D model to the intraoperative laparoscopic view. Clinically available state-of-the-art commercial systems use manual point-based rigid registration [170], while recent research works focus on either rigid [127, 178] or deformable [30, 56, 104, 161] registration techniques requiring di erent amounts of manual interactions and computations on the preoperative data. In contrast, the framework detailed in this Chapter formulates the deformable registration as a shape completion problem that does not rely on patient speci c preoperative computations.

Even though the underlying techniques are di erent, the common application and presence of an optimisation procedure make the proposed method similar to registration. Since the proposed method has the potential to become a precursor to a new approach for registration, in Sec. 3.3 it is directly compared with a rigid registration algorithm (Go-ICP [213]) that aligns two point clouds. This algorithm was successfully used for laparoscopic liver applications in [127], where the preoperative model was registered onto the intraoperative point cloud obtained using an unsupervised neural network for depth estimation. The proposed method is similar, but relying on a manual interaction it is also able to predict a deformed model that better ts the point cloud. Other methods, such as [104, 161] attempt the registration of preoperative models directly on the intraoperative images requiring manual image annotations. Even though they still show high errors in areas not visible from the camera, these

methods showed extremely good performances in coping with deformations. Both use biomechanical models to simulate deformations, and [161] requires an additional preoperative step where multiple possible patient-specifc simulations have to be performed.

The most similar works to the one proposed in this chapter (and in particular in Sec. 3.2) are [1] and [121]. [1] leverages a voxel-based conditional VAE to complete missing segments of bone and plan jaw reconstructive surgical procedures. However, the anatomical structures are not deformable, the missing segments are small compared to the complete shape, and the solution is constrained by the remaining healthy tissues. Liver shape completion is more ill-posed because the liver is highly deformable and the missing parts are much larger than the partial intraoperative shape. In addition, as mentioned in Sec. 2.1, voxel-based representations of shapes are ine cient volume representations that struggle to achieve high resolutions and to handle deformations. Therefore, the proposed method represents 3D shapes as Riemannian manifolds discretised into meshes and uses GDL techniques to process these data. In particular, the proposed method adapts [121] to achieve shape completion in laparoscopic liver surgery by i() overcoming the shortage of data; (ii) compensating the lack of point correspondences between partial and complete shapes; i(i) rede ning the VAE training loss to deal with non-uniformly sampled meshes; andiv() leveraging preoperative data for the initialisation.

The optimisation process for shape completion, makes the methodology suitable for registration, but there are a few key steps that need innovating. Interestingly, this is the rst attempt to bring GDL methods operating on meshes in to computer assisted interventions.

3.2 Methods

The proposed method (Fig. 3.1) estimates the complete mesh of a liver given a partial point cloud of its surface. A graph convolutional VAE is trained to generate complete shapes (Sec. 3.2.1) and a distinct optimisation procedure

Figure 3.1: Schematic overview of the proposed intraoperative shape completion method. Top: a VAE ($\{E, G\}$) is trained on complete preoperative meshes of the liver. Bottom: the shape completion starts with a manual selection S on the vertices of the preoperative mesh X. The latent representation obtained encoding X is used to initialise z. The error between the selection on the generated mesh $X'_S = S\ G(z)$) and the partial intraoperative point cloud P is minimised optimising over z, R, t, which separately control the shape of new generated meshes, the rotation of and its translation respectively.

non-rigidly deforms them to t the partial point cloud (Sec. 3.2.2).

3.2.1 Shape Generator

As detailed in Sec. 2.1, a 3D mesh can be represented as a graph $\mathcal{M} = \{X, E, F\}$, where $X \in \mathbb{R}^{N \times 3}$ is its vertex embedding, $E \in \mathbb{N}^{\bar{N} \times 2}$ is the edge connectivity that de nes its topology, and $F \in \mathbb{N}^{\bar{N} \times 3}$ its triangular faces. Traditional convolutional operators, well suited for grid data such as images and voxelizations, are generally incompatible with the non-Euclidean domain of graphs. Following [121], the generative model is built with the Feature-Steered graph convolutions de ned in [200] and previously described in Sec. 2.3.1.

Like extensively described in Sec. 2.2.2, every VAE is made of an encoder-decoder pair $\{E, G\}$, where the decoder is used as a generative model and is usually referred to as generator. Both E and G are parametrised by neural networks whose building blocks are the Feature-Steered graph convolutions (Sec. 2.3.1). During training, the ELBO loss in Eq. 2.2 is maximised. In practice, this is equivalent to minimising a reconstruction loss L_{recon}) encouraging the output of the VAE to be as close as possible to its input and a regularisation term (L_{KL}) pushing the variational distribution towards the prior distribution $p(z)$, which is de ned as an isotropic unit Gaussian distribution. While we set

L_{KL} to be the KL divergence (Eq. 2.1), we de ne L_{recon} as a vertex-density-weighted mean-squared-errors loss. Let x_i be the i-th position vector (i.e. a feature vector of size 3) and x_i^0 its corresponding point in $X^0 = G(E(X))$. We have:

$$L_{recon} = \frac{1}{N} \sum_{i=0}^{N-1} \| x_i^0 \quad x_i \|_2^2 \quad \text{with} \quad / \quad \frac{1}{N_i} \sum_{j \in N_i} \| x_i \quad x_j \|_2^2 \qquad (3.1)$$

where is a vertex-wise weight that increases the contribution of the errors in low vertex-density regions, thus preventing the generated mesh from t-ting only densely sampled areas. The total loss is then computed as linear combination of the two terms: $L_{tot} = L_{recon} + \quad L_{KL}$.

3.2.1.1 Data Preparation

Though the chosen graph convolution was proven e ective also on datasets with di erent graph topologies [200], the liver dataset is remeshed in order to have the same topology and known point correspondences across all the preoperative meshes and all the generated shapes. This accelerates and eases the training procedure, making possible the formulation of a simple and computationally-e cient loss function (Eq. 3.1). In addition, thanks to the consistent vertex indexing it is possible to easily perform the initial manual selection described in Sec. 3.2.2.

In order to consistently remesh all the livers in the dataset, an optimisation procedure iteratively deforms an ico-sphere with a prede ned topology and xed number of vertices to each liver. Following [203], the loss function used during the optimisation is given by $L_{remesh} = \ _0 L_{Ch} + \ _1 L_n + \ _2 L_L + \ _3 L_E$. $_{0;1;2;3}$ are the weights of each loss term L_{Ch} is the Chamfer distance that averages the distances between each point in a mesh and the closest point in the other mesh, and vice versa. Being s a point on the ico-sphere, and r a point on the raw livers, the Chamfer distance can be written as:

$$L_{Ch} = \sum_s \min_r \| s \quad r \|_2^2 + \sum_r \min_s \| s \quad r \|_2^2 :$$

L_n is the normal loss that requires the edge between a vertex and its neighbours to be perpendicular to the normal of the closest vertex in the ground truth. It can be formulated as:

$$L_n = \sum_{s} \sum_{r = \arg\min_r (\|s - r\|_2^2)} \langle s - k, n_r \rangle^2_2, \qquad \text{s.t. } k \in N(s),$$

where s is identified during the computation of the Chamfer distance as the closest vertex of the deforming ico-sphere to the points on the surface of the raw liver. k is a neighbour of s, n_r is the normal of point r, and $\langle \cdot, \cdot \rangle$ is the inner product. L_L is the mesh Laplacian regularisation loss based on the Tutte Laplacian formulation (Eq. 2.14). In its matrix form, it is computed as:

$$L_L = \frac{1}{N} \| TS \|_F^2,$$

where S are the vertex positions of the deforming ico-sphere with N vertices and F indicates the Frobenius norm. This loss encourages smoothness and helps avoiding self-intersections. Finally, L_e is an edge regularisation that reduces flying vertices by penalising long edges and that is computed as:

$$L_e = \sum_{s} \sum_{k \in N(s)} \| s - k \|_2^2.$$

3.2.1.2 Spectral Augmentation

The small size of liver datasets makes it difficult to train a generative model that can generalise to new shapes. Simple shape augmentation techniques such as random rotations, translations and scalings can be used to augment the dataset, but shapes are not deformed and the performance gain is therefore limited. Instead of attempting an augmentation in the spatial domain the data augmentation is performed by operating in the frequency domain. Although shape deformation in the frequency domain is a known concept in the literature [180], the proposed approach simplifies and randomises the spectral deformation making it suitable for data augmentation. Computing the eigenvalue

decomposition of the Kircho Laplacian (Eq. 2.13), $K = U U^T$, produces a set of orthonormal eigenvectors (columns of U) which are the Fourier bases of the mesh, and a series of eigenvalues (diagonal values of Λ) that are its frequencies. The Fourier transform of the vertices can be computed as $\hat{X} = U^T X$ and the inverse Fourier transform as $X = U \hat{X}$ [49].

Using these operators, each mesh is transformed into its spectral domain, perturbed, and transformed back to the spatial domain. Hence, the spectral augmented mesh is computed as $X_{aug} = U \; U^T X$, where is a vector that randomly perturbs four mesh frequencies. In particular, the rst frequency is never modi ed because, playing the role of a direct current component [28], it would not deform the shape. One of the following three frequencies, responsible for low frequency variations similar to scalings along the three major axes of the mesh, is arbitrarily perturbed. The remaining three perturbations are applied to randomly selected higher frequencies with the e ect of a ecting the ne details of the shape.

It is worth noting that the remeshed data share the same topology, thus the set of orthonormal eigenvectors used to compute the direct and inverse Fourier transforms can be computed one time and then used for every mesh.

3.2.2 Shape Completion

This section illustrates how a complete shape is obtained from a partial intraoperative point cloud $P \in R^{P \times 3}$. In contrast to [121], known (or easily computable) point correspondences between intraoperative point clouds and the generated meshes are not available. This assumption is relaxed at the expense of the introduction of a manual step in the procedure. In fact, the surgeon is asked to roughly select from the preoperative 3D model $X \in X_c$ a region of interest that corresponds to the visible surface in the intraoperative image. To reduce computational time and increase robustness against the errors in manual region selection and varying vertex density in the region, a subset of the selected vertices is sampled with an iterative farthest point sampling [172], obtaining a selection operator S that gives sparser and uniformly

Figure 3.2: Schematic description of the intraoperative liver shape completion. Left: the shape completion starts with a manual selection S on the vertices of the preoperative mesh X. Centre: the latent representation obtained encoding X, $z_0 = E(X)$, is used to initialise z for the completion procedure. z_0 is then fed to the generator G, which reconstructs the preoperative shape used to begin the shape completion. Right: the error between the selection on the generated mesh ($X^0_S = S \quad G(z)$) and the partial intraoperative point cloud P is minimised optimising over $z; R; t$, which separately control the shape of new generated meshes, the rotation of P and its translation respectively.

sampled vertices (Fig. 3.2 Left). Since mesh topology consistency is guaranteed by construction, the selected vertices will always have the same indexing for each mesh $X^0 2 X_c$ generated with the model discussed in Sec. 3.2.1. The shape completion problem is thus formulated as nding the best latent variable z that generates a complete shape X^0 plausibly tting P (Fig. 3.2 Centre). Given $X^0_S = S \quad X^0 = S \quad G(z) \quad X^0$ the subset of selected and sampled vertices from a generated shape, the optimisation is formulated as:

$$\min_{z;R;t} L_{Ch} \quad S \quad G(z); \; RP + t \quad : \tag{3.2}$$

It is worth mentioning that not having point correspondences between P and X^0_S it is not possible to compute the rotation R and translation t in a closed form solution as in [121]. Therefore, they are iteratively updated alongside z in the same optimisation procedure. The gradient of the loss in Eq. 3.2 directly in uences R and t, but it needs to be back-propagated through the generator network G, without updating the network's weights, to update z. The completion procedure is initialised by centering P and X^0_S, and by setting an initial $z = z_0 = E(X)$, thus using as prior the latent representation of the preoperative mesh (Fig. 3.2 Right). The initialisation z_0 can be further re ned to z_0 by running a few iterations of a second optimisation

$z_0 \quad \arg\min_{z_0} \quad \max_i \, kx_i \quad x_i^0 k_2^2$. Finally, by adding to the latent initialisation a small Gaussian noise $\quad G(0; \quad)$ with $_{ii} \quad I_{ii}$, it is also possible to generate multiple complete shapes conditioned on the preoperative data and that plausibly t the intraoperative point cloud P. The hyperparameters used during shape completion are detailed in Sec. 3.3.2.

3.3 Results

The proprietary dataset collected for this work consists of 50 meshes of livers which were segmented and reconstructed from preoperative CT scans of di erent patients. The segmentation and initial mesh generation was performed by Visible Patient. 45 meshes were used to train the VAE, while the remaining 5 meshes were used as a test set to evaluate the network, data augmentation, and shape completion. Given the limited size of the dataset, to not bias results toward the test set, hyperparameters were tuned on the training set. The study was approved by the local research ethics committee (Ref: 14/LO/1264) and written consent obtained from each patient.

The remeshing is performed by deforming an ico-sphere with 2564 vertices. For this, and all the other optimisations described in this paper, the Adam optimiser [97] is used. All the raw liver shapes are remeshed with 500 iterations at a xed learning rate of $lr = 5e^{\ 3}$. The weights of the loss function L_{remesh} are $_0 = 5$, $_1 = 0{:}2$, $_2 = 0{:}3$, and $_3 = 15$.

The VAE is built using $M = 8$ weight matrices, batch size 20 and latent size 128. LeakyReLU and batch normalisation are used after every layer (see architecture details in the appendix Sec. A.1). The network is implemented in PyTorch Geometric [60] and trained for 200 epochs with $lr = 1e^{\ 3}$ and a KL divergence weight $= 1e^{\ 6}$. The training was performed on an NVIDIA Quadro P5000 and took approximately 9 hours.

The reconstruction performance of the VAE is evaluated with and without data augmentation while xing the number of iterations. Applying the spectral augmentation (Fig. 3.3A) 100 new meshes are generated for each model in the

Figure 3.3: Augmentation and Qualitative Results A: e ects of the spectral augmentation where a real liver (green) is subject to two di erent random augmentation (black). B: laparoscopic image and comparison between the proposed shape completion (blue) and the Go-ICP registration (black). The intraoperative point cloud is shown in red and the selected point in blue. The contours of the silhouettes are overlaid also on the laparoscopic image.

training set, thus obtaining 4500 models. An additional online data augmentation composed of random rotations, scalings, and translations is applied. A mean-squared testing error of (0.28 ± 0.04) mm is obtained when both augmentations are applied, of (0.45 ± 0.18) mm with the online augmentation alone, of (0.50 ± 0.03) mm with the spectral augmentation only, and of (0.92 ± 0.22) mm without any augmentation. The evaluation of the computational cost associated to the spectral augmentation reported that when it is performed by computing the Fourier operators for each mesh it takes 4.05 ± 0.05 seconds per mesh, while, when the operators are precomputed (Sec. 3.2.1) the computational time is reduced by one order of magnitude to 0.05 ± 0.01 seconds per mesh.

3.3.1 Qualitative Results

Given the lack of intraoperative 3D ground truths for registration in laparoscopic liver surgery, the evaluation of the proposed method on real data is purely qualitative. In a real operative scenario the computation of a dense and reliable point cloud is still a major challenge. An o -the-shelf depth reconstruction network [35] is used to obtain P from recti ed images of a calibrated laparoscope. Given the predicted depth map and a manual segmentation of the liver, P is computed rst, then the complete shape X^0 (Fig. 3.3B) is estimated.

3.3.2 Quantitative Results

The quantitative assessment of the shape completion is performed on synthetic data. The five meshes in the test set are manually deformed, trying to reproduce intraoperative liver deformations similar to those expected in a laparoscopic procedure and characterised in [80]. To obtain intraoperative partial point clouds, the surface of the deformed models is sampled with vertex selections on three regions: entire front surface P_F, left lobe P_L, and right lobe P_R. Each deformed model is considered the intraoperative ground truth X_{GT} to be inferred given a partial point cloud. To maintain a higher P density, X_{GT} is not remeshed. Eq. 3.2 is optimised for 100 iterations using Adam with a different learning rate for each term. To encourage the optimisation over z and thus the generation of more diverse, progressively deformed meshes X^0, the learning rate responsible of updating the latent is set to $lr_z = 5e^{-2}$. The learning rates for R and t are empirically set to $lr_R = 1e^{-2}$ and $lr_t = 5e^{-5}$. Rotations are regressed faster because the two point clouds were initially centred.

Figure 3.4: Quantitative Results. Rows: results for intraoperative point clouds of front surface P_F, right lobe P_R, and left lobe P_L. Columns: A) front and back view of intraoperative point cloud P (red), intraoperative ground truth X_{GT} (green), and prediction X^0 (blue). B) Comparison of vertex-wise distances (mm) for the proposed method (blue) and Go-ICP (pink). Errors are separately computed in the invisible parts of the liver and in the visible parts, which are within the camera's field of view. C) Front and back view of X^0. Colours represent the algorithm with a smaller vertex error.

In case z_0 is further re ned to z_0, the same learning rate $r_z = 5e^{\ 2}$ is used for 20 iterations. In case multiple complete shape proposals are desired, Eq. 3.2 can be generalised to process batches with a re ned initialisation perturbed by with $_{ii} = \frac{1}{10}$.

The shape completion is evaluated for each partial shape without perturbation. Since the procedure requires a manual step currently performed with a lasso selection that might a ect results, the evaluation is repeated 3 times, for a total of 45 experiments. Selections could be re ned and took approximately one minute each. The proposed method is compared with the rigid registration using Go-ICP [127] which has comparable computational time. The lack of point correspondences between X_{GT} and X^0 does not make possible to evaluate the proposed method using mean-squared errors. Therefore, a variation of L_{Ch} computing vertex-wise errors on X^0 is de ned. For each vertex of one mesh the distance to the closest point on the other mesh is computed. All the distances are assigned to the vertices of X^0 from which they were computed and are locally averaged. Results are reported in Fig. 3.4.

3.4 Discussion

While this work is about shape completion, it could become an alternative to registration or provide a better initialisation for such algorithms. From Fig. 3.3 it can be observed that the proposed method seems to t better the point cloud especially on the left lobe. The lack of a ground truth makes impossible to draw further conclusions from this result. However, observing Fig. 3.4 (columns B and C), it can be concluded that the proposed method outperforms Go-ICP in visible areas, and, despite performing worse in invisible areas, it predicts a realistic looking model of the liver. Of particular importance is the improvement over visible areas, because these regions are the only ones in the narrow eld of interest of the surgeon, where an accurate deformation is required. Since the manual selection of the visible area on the preoperative model a ects the quality of the results, future work not only

should aim at quantifying the uncertainty involved in the manual interaction, but also at avoiding it by predicting point correspondences between partial and complete shapes. In addition, the use of biomechanical constraints for deformation could reduce errors in invisible areas. In fact, the unconstrained deformations operated by the proposed method through the optimisation of z, despite generating plausible livers tting the partial intraoperative point cloud, often downscale invisible areas. Even though the proposed method can propose multiple solutions (Sec. 3.2.2), identifying the correct complete shape is essential to improve the method and outperform Go-ICP everywhere. Thus, the introduction of biomechanical constraints shall be researched while avoiding any patient speci c training or simulation. To introduce these constrains locally, the generative model would need to have control over the generation of speci c parts of the liver. This can be achieved by further disentangling the factors of variation optimised during shape completion. Since datasets of liver shapes are limited in size and livers do not have peculiar local shape attributes, the disentanglement methods proposed in this thesis are developed relying on digital human shapes where there is no paucity of data and shapes parts are easily distinguishable. Nevertheless, the methods proposed in the following chapters could be applied also to liver shapes and explored in conjunction with the introduction of biomechanical constraints.

Chapter 4

Swap Disentangled Variational Autoencoder

Learning a disentangled, interpretable, and structured latent representation in generative models of 3D human shapes is still an open problem. The problem is particularly acute when control over local shape properties is required. Experimental results conducted on 3D meshes show that state-of-the-art methods for latent disentanglement are not able to disentangle identity attributes of faces and bodies (e.g., mouth, nose, arms, chest, etc.). The intuitive yet e ective self-supervised approach described in this chapter, called SD-VAE, can be used to train a 3D shape VAE which encourages a disentangled latent representation of identity attributes. Curating the mini-batch generation by swapping arbitrary attributes across di erent shapes allows to de ne a loss function leveraging known di erences and similarities in the latent representations. The proposed method properly decouples the generation of such attributes while maintaining good representation and reconstruction capabilities.

A paper on this work has been published as [62]: Simone Foti, Bongjin Koo, Danail Stoyanov, Matthew J. Clarkson. 3D Shape Variational Autoencoder Latent Disentanglement via Mini-Batch Feature Swapping for Bodies and Faces *. Proceedings of the IEEE/CVF Conference on

* Although they can be used interchangeably, while in the published paper we refer to local shape parts as \features", in this thesis we refer to them as \attributes" to improve consistency with the following chapters.

Computer Vision and Pattern Recognition (CVPR), pp. 18730-18739. 2022
© 2022 IEEE.

In this Chapter, after a brief introduction (Sec. 4.1), the proposed method is described (Sec. 4.2). The method is then thoroughly evaluated and compared in the experiments section (Sec. 4.3). A discussion on the strengths of the method as well as its limitations and future research directions concludes the chapter (Sec. 4.4).

4.1 Introduction

As detailed in Sec. 1.1, the generation of 3D human faces and bodies is a complex task with multiple potential applications. Currently, the generation procedure is either manually performed by highly skilled artists or it involves semi-automated avatar design tools. Even though these tools greatly simplify the design process, they are usually limited in exibility because of the intrinsic constraints of the underlying generative models (Sec. 1.1). These models all share one particular issue: the creation of local attributes is di cult or even impossible. In fact, not only do generative coe cients (or latent variables) lack any semantic meaning, but they also create global changes in the output shape. For this reason, this chapter focuses on the problem of 3D shape creation by enforcing disentanglement among sets of generative coe cients controlling the identity of a character.

Unlike the latent disentangled models capable of controlling local shape properties introduced in Sec. 2.4, the SD-VAE here proposed relies on a single VAE, which is trained by curating the mini-batch generation procedure and with an additional loss. The intuition behind this method is that when swapping attributes (e.g. nose, ears, legs, arms, etc.) across the input data in a controlled manner (Fig. 4.1,Left), not only it is possible to know a priori which shapes within a mini-batch have (do not have) the same attribute, but also to know which are (are not) created from the same face (body). These di erences and similarities across shapes should be captured in the latent representation.

Therefore, assuming that di erent subsets of latent variables correspond to different attributes, it is possible to leverage the structure of the input batch to encourage a more disentangled, interpretable, and structured representation.

With the objective of building a model capable of generating 3D meshes, a VAE architecture is built by extending [69](see Sec. 2.3.2). This state-of-the-art model proved to be fast and capable of better capturing non-linear representations of 3D meshes, while leveraging very intuitive convolutional operators characterised by a reduced number of parameters. Nonetheless, the network choice is arbitrary and the method is expected to be working also with other network con gurations and operators. Even though meshes are the primary data structures, it is also worth noting that, by providing semantic segmentations of the di erent features, SD-VAE is applicable to voxel- or point-cloud-based generative models. The generality of the proposed method is particularly important in the current geometric deep learning eld, where de nitions of 3D convolutions and pooling operators are still an open problem.

To summarise, the key contributions of the proposed approach are: (i) the de nition of a new mini-batching procedure based on attribute swapping, (ii) the introduction of a novel loss function capable of leveraging shape di erences and similarities within each mini-batch, and (iii) the consequent creation of a 3D-VAE capable of generating 3D meshes from a more interpretable and structured latent representation.

4.2 Method

The proposed method (Fig. 4.1) provides more interpretable and structured latent representations for self-supervised 3D generative models. This is achieved by training a mesh-convolutional variational autoencoder (Sec. 4.2.1) with a mini-batch controlled attribute swapping procedure and a latent consistency loss (Sec. 4.2.2).

Figure 4.1: Schematic description of the SD-VAE, a 3D-VAE where each subset of latent variables controls a shape attribute. Left: at the core of the proposed method lies a curated mini-batching procedure such that a mini-batch of vertices (X_{ij}) is created by swapping an arbitrary attribute across di erent 3D shapes. Notice that attributes from the same identity have the same colour. Centre: a 3D-VAE $(f E; G g)$ encodes X_{ij} in its latent representations $z_{ij} = (z^f_{ij} j z^c_{ij})$, which are subsequently decoded into X^0_{ij}. In this case f corresponds to the nose. Therefore, thanks to the proposed latent consistency loss, the latent subsets z^f_{ij} control the shape of the nose and z^c_{ij} control the shape of the rest of the face. Right: visual representation of all the di erent mesh attributes for which it is sought to obtain a disentangled latent representation. © 2022 IEEE.

4.2.1 Mesh Variational Autoencoder

Following the convention introduced in Sec. 2.1, a manifold triangle mesh is de- ned as $M = f X; E; F g$, where $X 2 R^{N 3}$ is its vertex embedding, $E 2 N^{" 2}$ is the edge connectivity that de nes its topology, and $F 2 N 3$ are its triangular faces. Assuming that meshes share the same topology across the entire dataset, E and F are constant and meshes di er from one another only for the position of their vertices, which are assumed to be consistently aligned, scaled, and with point-wise correspondences. Since traditional convolutional operators are not compatible with the non-Euclidean nature of meshes, the proposed generative model is built with the simple yet e cient approach de ned in [69]. Convolution operators are thus de ned as learnable functions over pre-computed dilated spiral sequences [69]. Pooling and un-pooling operators are de ned as sparse matrix multiplications with pre-computed transformations that are obtained with a quadric sampling procedure [69, 174] (see Sec. 2.3.2).

The proposed 3D-VAE is built as an encoder-decoder pair (Fig. 4.1, Cen-

tre), where the decoder is used as a generative model and is referred to as generator. Throughout the entire network, each spiral++ convolutional layer is followed by an ELU activation function. However, in E convolutions are interleaved with pooling layers and in G by un-pooling layers. There are also three fully connected layers: two of them are the last layers of E predicting the mean and the diagonal covariance of the variational distribution, the other is the first layer of G and transforms $z \in Z$ back into a low-dimensional mesh that can be processed by mesh convolutions (more details are provided in the appendix Sec. A.1).

Like detailed in Sec. 2.2.2, during training the objective is to maximise the ELBO loss. In practice, maximising the ELBO in Eq. 2.2 is equivalent to minimising the following loss:

$$L_{VAE} = L_R + \quad L_L + \quad L_{KL} \tag{4.1}$$

where and are weighting constants. $L_R = \frac{1}{N}\sum_{n=1}^{N} \|x_n^0 \quad x_n\|_2^2$ is the mean squared error between the input $x_n \in X$ and the corresponding output ($x_n^0 \in X^0 = G(E(X)) = G(z)$) vertices. This reconstruction loss encourages the output of the VAE to be as close as possible to its input. $L_{KL} = D_{KL}[q(z|X)||p(z)]$ is the KL-divergence (Eq. 2.1) pushing the variational distribution $q(z|X)$ towards the prior distribution $p(z)$, which is defined as an isotropic unit Gaussian distribution. Finally, L_L is a smoothing term, not conventionally present in Eq. 2.2 and based on the Tutte Laplacian [150] that is computed on the output vertices as:

$$L_L = \frac{1}{N}\sum_{n=1}^{N} \|\ _n\|_2 \quad \text{with} \quad \ _n = \frac{1}{|N_n|}\sum_{e \in N_n} x_e^0 \quad x_n^0$$

where $\ _n$ is the Laplacian of the n-th output vertex, and N_n the set of its neighbouring vertices with cardinality $|N_n|$. L_L is efficiently computed by relying on matrix operators like in Sec 3.2.2. Concretely, $= [\ _1, \ldots, \ _N]^T = TX^0$, where T is the Tutte Laplacian operator defined in Eq. 2.14. Note that

Figure 4.2: Examples of attribute swapping for di erent attributes and di erent subjects. © 2022 IEEE.

vertices are normalised by subtracting the per-vertex mean of the training set and dividing the result by the per-vertex standard deviation of the training set, thus the losses in Eq. 4.1 are computed on normalised vertices. Also, all loss terms are reduced across mini-batches with a mean reduction.

4.2.2 Mini-Batch Attribute Swapping and Latent Consistency Loss

The objective is to create a generative model where vertices corresponding to speci c mesh attributes are controlled by a prede ned set of latent variables. To accomplish this, the rst step requires to de ne F arbitrary mesh attributes on a mesh template (Fig. 4.1, Right) by manually colouring mesh vertices. Since vertices have point-wise correspondences (Sec. 4.2.1), attributes can be easily identi ed for every other mesh in the dataset without manually segmenting them. This makes possible to swap attributes from one mesh to another by replacing the vertices corresponding to the selected attribute (Fig. 4.2).

Attribute swapping is at the core of SD-VAE and it facilitates the curation of the mini-batch generation in order to properly shape and constrain the latent representation of each mesh. Each mini-batch of size B can be thought of as a squared matrix of size $\sqrt{B} \times \sqrt{B}$, where each element X_{ij} is the vertex embedding of a di erent mesh. As it can be seen from Fig. 4.1 (Left), while elements on the diagonal of this matrix are loaded from the dataset, the remaining elements are created online by swapping attributes. Every time a mini-batch is created, an attribute is randomly selected and swapped. Therefore, each

row of the matrix contains the same mesh with di erent attributes, while each column contains di erent meshes with the same attribute. Interestingly, the naive implementation of the attribute swapping causes visible surface discontinuities in most input meshes (Fig. 4.2), but discontinuities are not present in reconstructed meshes thanks to the Laplacian regulariser in Eq. 4.1.

Obviously, when a mini-batch is encoded a batched latent is computed. As shown in Fig. 4.1 (Centre), for each X_{ij} there is a corresponding $z_{ij} \quad E(X_{ij})$ which is evenly split in F subsets of latent variables, one for each mesh attribute ($z_{ij} = \{z^{l}_{ij}\}_{l=1}^{F}$). Note that even though every latent subset z^{l}_{ij} has the same number of variables, uneven splits are also admissible.

Every time a mini-batch is created by swapping an attribute f, it is possible to de ne $z_{ij} = (z^{f}_{ij} | z^{c}_{ij})$. z^{f}_{ij} is the subset of latent variables controlling the attribute swapped across the current mini-batch. z^{c}_{ij} is the part that controls everything else and is de ned as $z^{c}_{ij} = \{z^{l}_{ij}\}_{l=1}^{F} \cap \{z^{f}_{ij}\}$. Inspired by both triplet losses and [183], and thanks to the curated mini-batching, di erences and similarities in the latent representation of the di erent X_{ij} can be enforced by requiring matched z^{l}_{ij} pairs to have a distance in latent space that is smaller by a margin, , than the distance for unmatched pairs. While traversing the diagonal of the mini-batch latent matrix, all the elements on the row containing the diagonal element z_{ss} are compared with those in the column containing z_{ss} ($\forall s \in \{1, \ldots, \sqrt{B}\}$). When considering z^{f}_{ij} latent similarities are enforced across columns and latent di erences across rows by evaluating: $\|z^{f}_{ps} \quad z^{f}_{qs}\|_2^2 + \quad {}_1 \quad \| z^{f}_{sp} \quad z^{f}_{sq}\|_2^2, \forall s, p, q \in \{1, \ldots, \sqrt{B}\}$ with $p \neq q$. This is justi ed by the fact that elements in X_{ij} have the same mesh attribute across columns and di erent mesh attributes across rows. Vice versa, when considering z^{c}_{ij}, which controls all the other mesh attributes for the current mini-batch, similarities are enforced row-wise and di erences column-wise by evaluating: $\|z^{c}_{sp} \quad z^{c}_{sq}\|_2^2 + \quad {}_2 \quad \| z^{c}_{ps} \quad z^{c}_{qs}\|_2^2, \forall s, p, q \in \{1, \ldots, \sqrt{B}\}$ with $p \neq q$. The proposed

latent consistency loss is thus de ned as:

$$L_c = \sum_{\substack{s;p;q=1 \\ p \neq q}}^{\overline{B}} \max\left(0; k z^f_{ps} \quad z^f_{qs} k_2^2 \quad k\, z^f_{sp} \quad z^f_{sq} k_2^2 + \quad {}_1\right) + \\ + \max\left(0; k z^c_{sp} \quad z^c_{sq} k_2^2 \quad k\, z^c_{ps} \quad z^c_{qs} k_2^2 + \quad {}_2\right) \tag{4.2}$$

where $= \frac{1}{\overline{B}\,(\overline{B} \quad \overline{B})}$ is a batch normalisation term that considers all the latent distances comparisons performed while computing L_c. Combining Eq. 4.1 with Eq. 4.2 and said $2\ R$ a weighting coe cient, it is nally possible to formulate the total loss as:

$$L = L_{VAE} + \quad L_c = L_R + \quad L_L + \quad L_{KL} + \quad L_c \tag{4.3}$$

4.3 Experiments

4.3.1 Datasets

The main objective of this work is to train a generative model capable of generating di erent identities from a set of attribute-disentangled latent variables. For the experiments, datasets containing as many subjects as possible in a neutral expression are needed. However, most open source datasets for 3D shapes of faces, bodies, or animals contain only a limited number of subjects captured in di erent expressions or poses (e.g. Mpi -Dyna [169], Smpl [125], Surreal [199], Coma [174], Smal [228], etc.). For this reason, this work relies on two linear models that were built using a conspicuous number of subjects and that are released for non-commercial scienti c research purposes: Uhm [168] and Star [158] (Sec. 2.2). From these models, 10,000 meshes are randomly sampled and subsequently used to create one dataset for faces and one for bodies. 90% of the data are used for training, 5% for validation, and 5% for testing.

4.3.2 Implementation Details

All networks were implemented in PyTorch and trained for 40 epochs using the Adam optimiser [98] with a fixed learning rate of $lr = 1e^{-4}$ and mini-batch size $B = 16$ (note that the attribute swapping is applied to SD-VAE only). Spiral convolutions had spiral length of 9 and spiral dilation of 1. The last convolutional layer of E and the first of G had 64 features while all the others 32. The sampling factors used during the quadric sampling for the creation of the up- and down-sampling transformation matrices were set to 4. Since the two datasets have a significantly different number of vertices ($N_{faces} = 71,928$ and $N_{bodies} = 6,890$), networks operating on faces have 4 convolutional layers interleaved with sampling operators in both E and G, while networks operating on bodies have only 3. For the same reason latent sizes are different: 60 variables for faces and 33 for bodies. Considering that the face template was segmented in 12 regions and the body template in 11, each z_i has 5 variables for faces and 3 for bodies (more details on the architecture are reported in the appendix Sec. A.1). The weight of the Laplacian regulariser was set to $= 1$, while the latent consistency weight was $= 0.5$ for faces and $= 1$ for bodies. $_1$ and $_2$ were set to $_1 = {}_2 = 0.5$ for both datasets. Training was performed on a single Nvidia Quadro P5000 for faces and on an Nvidia GeForce GTX 1050Ti for bodies. Approximately 120 experiments were run in 25 GPU days.

4.3.3 Comparison with Other Methods

The proposed SD-VAE is compared with other self-supervised methods based on encoder-decoder pairs. For a fair comparison, all methods share the same underlying architecture, which is referred to as VAE and which is already detailed in Sec. 4.2.1. Consistently with the current literature [174, 121, 61, 216], it was found that the weight coefficient ( ) on the KL divergence in VAEs for meshes is smaller than the one used for images. In fact, with 1 the VAE is not able to reconstruct the data. Thus, results are reported on VAEs with $2 \{1e^{-2}, 1e^{-4}\}$. It is worth noting that the discrepancy between meshes and images does not allow to define a -VAE with the same criteria used in