

NRIC vs Vertex coordinates

Evaluating the benefits of Rotation-Invariant Coordinates for Machine Learning on 3D Meshes

by

Sebastian-Alexandru Manda

to obtain the degree of Master of Computer Science
at the Delft University of Technology,
to be defended publicly on Thursday July 9, 2026 at 12:00 PM.

Student number:	5259096
Project duration:	November 20, 2025 – July 9, 2026
Thesis committee:	Prof. Klaus Hildebrandt, TU Delft, supervisor Jackson Campolattaro, TU Delft, co-supervisor Assistant Prof. David Tax, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Three-dimensional meshes are central to computer graphics, animation and simulation, where they are almost always stored as a set of vertices connected by faces. While this is the standard representation for meshes, it is difficult to work with in machine learning applications, since raw vertex coordinates only describe the absolute position of the geometry and are easily disrupted by any rigid body transformation. Nonlinear Rotation-Invariant Coordinates, or NRIC, offer a different starting point, describing a mesh through the lengths of its edges and the dihedral angles between two faces intersecting at each edge. The advantage of NRIC is twofold. It is invariant to all rigid body motions, which isolates the geometric structure of the mesh from its position and orientation, and it tends to make deformations between mesh states more linear and structured.

This thesis sets out to compare NRIC and raw vertex coordinates as inputs for machine learning, in order to determine whether NRIC can be a more beneficial starting point. The comparison is carried out across two tasks, mesh sequence compression through overfitting and latent space extraction with a variational autoencoder, each evaluated on both representations. The architectures are kept as identical as possible between the two inputs, so that any difference can be attributed to the representation itself rather than the application.

The experimental results show that NRIC is not always an improvement over raw vertices. For compression, models trained on NRIC lose quality in their predictions, from errors which are amplified by the reconstruction steps that return the NRIC features to a 3D mesh for evaluation. With latent space learning, however, NRIC shows clear promise, encoding collections of meshes in a more organized manner that usually improves over the standard 3D representation. In both applications, NRIC carries trade-offs, both in the overhead it adds during training and evaluation and in its tendency to benefit only certain types of meshes and settings. NRIC is therefore not the right choice for most applications, but that is not to say it cannot excel in others, where its benefits outweigh the costs.

Contents

Abstract	ii
1 Introduction	1
1.1 Context and Motivation	1
1.2 Research Problem and Scope	1
1.3 Research Questions	2
1.4 Contributions	2
1.5 Structure of the Thesis	3
2 Background	4
2.1 Conventional Mesh Representation	4
2.2 Nonlinear Rotation-Invariant Coordinates	4
2.3 Euclidean and Non-Euclidean Spaces	5
2.4 Latent Spaces	6
3 Related Work	7
3.1 Machine Learning on Conventional Mesh Representations	7
3.2 Rotation-Invariant and Intrinsic Representations	7
3.3 Latent Space Learning for Mesh Synthesis	8
4 Methodology	10
4.1 Mesh sequence compression via overfitting	10
4.1.1 Network Architecture	10
4.1.2 Fourier Time Encoding	10
4.1.3 Vertex Coordinate Approach	11
4.1.4 NRIC Approach	12
4.2 Variational Mesh Encoding	13
4.2.1 Network Architecture	14
4.2.2 Preprocessing	15
4.2.3 Training Objective	16
4.2.4 Free Bits	16
5 Experimental Results	17
5.1 GOAST Python bindings	17
5.2 Mesh Sequence Compression	19
5.2.1 Experimental Setup	19
5.2.2 Direct Time-to-Feature Mapping	20
5.2.3 Origin-Relative Feature Encoding	21
5.2.4 Interpolating Frames from Compressed Encoding	23
5.2.5 Feature-to-Feature Mapping	24
5.2.6 Discussion	25
5.3 Latent Space and Mesh Synthesis	26
5.3.1 Experimental Setup	27
5.3.2 Reconstruction Quality	28
5.3.3 Dimension Analysis	28
5.3.4 Latent-to-Geometry Correlation	29
5.3.5 Latent Space Traversal	30
5.3.6 Interpolation and Generation	32
5.3.7 Class Separation	33
5.3.8 Discussion	35

6 Conclusion	37
7 Reflection	40
7.1 Limitations	40
7.2 Wider Context and Implications	41
7.3 Use of AI	42
Bibliography	43
A Appendix A	45

Introduction

1.1. Context and Motivation

Three-dimensional meshes form the backbone of modern computer graphics, animation, and simulation. The conventional representation of these meshes, a set of vertices positioned in 3D space and connected by faces, has remained the standard since the early days of modern computers. It is a stable and well-understood approach, supported by a vast ecosystem of tools, libraries and software. However, when it comes to machine learning trained on 3D mesh data, this format has proven difficult to work with, requiring increasingly complex methods to obtain meaningful data.

The main problem is that raw vertex coordinates do not really capture the structure and geometry of a mesh. A small rotation, for example, leaves the shape almost the same while changing every vertex coordinate by a large amount. As a result, an ever-growing collection of feature preprocessing and extraction techniques has been developed to close this gap. This ranges from surface normals and curvature descriptors [7] to spectral embeddings [2] and learned latent representations [13]. Each new application tends to demand its own tailored pipeline to give models access to geometric information that is simply not legible from vertices and faces alone. Methods such as variational mesh autoencoders [22] and latent diffusion approaches for motion synthesis [4] demonstrate the complexity these pipelines have reached, achieving impressive results in tasks ranging from shape compression [14] to novel mesh synthesis. However, what they have in common is the dependence on significant computation added to the base mesh representation before learning can meaningfully begin.

Nonlinear Rotation-Invariant Coordinates (NRIC), introduced by Sassen et al. [20], take a very different approach to representing the same mesh. Instead of using the positions of each vertex in 3D space, NRIC describes the mesh in terms of its edge lengths and dihedral angles between neighbouring faces. By construction, this representation is invariant under rigid-body transformations such as rotations and translations, so the same shape always maps to the same coordinates regardless of its orientation or position in space. Besides invariance, we expect that deformations between mesh states become more linear and structured in NRIC space, even if they look highly nonlinear when you follow the vertex paths through 3D space. If that holds, NRIC could be a more natural and meaningful medium for machine learning on meshes, one in which the useful geometric information is already part of the base representation, reducing the need for the heavy feature engineering that conventional coordinates require.

1.2. Research Problem and Scope

This thesis looks at whether NRIC is a better medium for machine learning on 3D meshes than the conventional vertex coordinate system. The main premise is that, if deformations are more linearly structured in NRIC space, then models operating in that space should require less capacity, less data or less extra feature engineering to match or beat models trained on raw vertex coordinates.

To look into this, we use two different machine learning tasks on mesh collections and sequences. The first is sequence compression through overfitting, where a small network is trained on a single animation sequence with the goal of encoding it as accurately as possible inside the model's weights. The second is latent space learning, where a model learns to embed meshes into a low-dimensional

continuous space from which new, hopefully plausible mesh geometries can be generated. The two tasks test different properties of each coordinate system. Compression checks how efficiently a neural network can encode each representation, while latent space learning checks how well the geometry of each representation supports smooth interpolation and generalisation.

We have intentionally focused the scope of our work. Our aim is not to create a general-purpose mesh learning framework or to claim that NRIC is a universally better choice. Instead, we keep the machine learning architectures and training procedures as consistent as possible across both representations, so that any difference in their results comes from the coordinate system itself rather than the model. Our experiments use datasets of varying complexity, starting with small custom sequences with low vertex counts and progressing to well-known benchmarks for broader evaluation [1, 18]. By comparing the two representations directly in these controlled settings, this thesis aims to give concrete and reproducible evidence of the practical trade-offs involved in choosing one coordinate system over the other, as well as to encourage more research in this direction.

1.3. Research Questions

The central question this thesis aims to answer is the following.

Are the Nonlinear Rotation-Invariant Coordinates (NRIC) more effective for machine learning applications when compared to conventional raw vertex coordinates?

This question is intentionally broad, as the answer may differ depending on the type of machine learning task being performed. To provide a more structured investigation, we break it down into a set of more focused sub-questions that each address a specific aspect of the comparison:

1. Where NRIC underperforms, does the error originate in the network's prediction of the intrinsic features or in the geometric reconstruction step that follows, and how does this balance change as mesh resolution increases?
2. Can a compression model overfitted to a single sequence keep enough temporal structure to reconstruct unseen intermediate frames, and is this limited by the compression approach itself rather than the choice of coordinate system?
3. Does a latent space learned from NRIC-encoded meshes organize mesh collections more meaningfully than one learned from raw vertex coordinates?
4. How do the two latent spaces differ when synthesizing and navigating new geometry, and does NRIC trade fidelity for guaranteed geometric validity?
5. Do NRIC's advantages and disadvantages relative to vertex coordinates generalize across machine learning tasks, or are they specific to the task at hand?

1.4. Contributions

This thesis makes three primary contributions. The first is a set of Python bindings for the GOAST library. The only existing implementation for extracting NRIC features from meshes and reconstructing these meshes from NRIC features resides in the C++ GOAST library, with no Python interface available. These bindings make NRIC usable within our Python-based pipeline, and they form the technical foundation that the rest of our implementation and experiments are built on.

The second contribution is a controlled, representation-neutral comparison we make between NRIC and raw vertex coordinates. Across both tasks and all experiments, we keep the architecture, hyperparameters, learning objectives, and training procedures as similar as possible between representations. This isolates the effect of the representation itself as the variable under comparison, rather than the differences introduced by the rest of the setup.

The third contribution is the experimental framework and analysis built on top of this comparison. For both mesh sequence compression and latent space extraction through a variational autoencoder, we provide thorough descriptions of the methods, datasets, and results, along with an analysis of how each representation affects model behaviour in each setting. This is intended to make it easier for future work to both replicate the results as well as expand on our study.

1.5. Structure of the Thesis

The remainder of the thesis is organized as follows. Chapter 2 covers the background needed for the rest of the work, including 3D meshes, their NRIC counterparts, and a few other topics that come up in future discussions. Chapter 3 introduces work related to NRIC along with state-of-the-art approaches for conventional vertex coordinates. Chapter 4 describes the methods used to run and evaluate the two tasks that answer the constructed questions. Chapter 5 presents the experimental results, first for the mesh sequence compression and then for latent space learning, and interprets what each set of results means for our comparison. Chapter 6 brings these findings together, addressing the proposed subquestions and the main research question posed in the introduction. Finally, Chapter 7 reflects on the work as a whole, analyzing design decisions, limitations and where this research fits in with other similar approaches. In addition, complete tables of the averaged mesh sequence compression experiments can be found in Appendix A.

2

Background

This chapter sets out the technical foundation needed for the rest of the thesis. We begin with the conventional representation of 3D meshes and their properties that matter for machine learning. We then turn to the NRIC representation and explain how it encodes the same geometric information in a different way, along with why that difference can be an advantage in certain applications.

2.1. Conventional Mesh Representation

A 3D mesh is a discrete representation of a surface in 3D space. In its conventional form it has two main components, vertices and faces. Each vertex is a point in 3D space, defined by three coordinates $(x, y, z) \in \mathbb{R}^3$. Faces connect groups of vertices to represent the surface, and they are most often triangles, with each face storing three vertex indices. Together, these two components give the conventional representation that is used to store and render meshes.

For meshes that share the same topology (the same number of vertices connected in the same way), the face structure remains identical across all instances. This is the case for both animation sequences, where a single mesh deforms over time while its connectivity stays fixed, and for collections of meshes representing different poses or individuals of the same category. In these settings, the geometric state of any given frame or instance is entirely determined by its vertex positions alone. The face matrix contributes no new information other than what is already implied by the shared topology. For this reason, in our experiments, we use only vertex positions as input to our models.

This means that from a machine learning perspective, a mesh sequence or collection can be reduced to a set of matrices in $\mathbb{R}^{n \times 3}$, where n is the number of vertices. While this is a compact and more convenient representation, it still has some limitations that will be discussed in the following sections. Some of these limitations stem from the fact that vertex positions are sensitive to rigid body transformations and are not a direct representation of the geometric form of the mesh.

2.2. Nonlinear Rotation-Invariant Coordinates

Nonlinear Rotation-Invariant Coordinates, introduced by Sassen et al. [20], provide an alternative encoding of mesh geometry that addresses several shortcomings of vertex coordinates. Instead of representing a mesh through the positions of its vertices in 3D space, NRIC encodes it through two sets of intrinsic geometric quantities: the lengths of all edges and the dihedral angles between pairs of adjacent triangular faces.

For a mesh with $|E|$ edges, the edge lengths are collected into a vector $l \in \mathbb{R}_{>0}^{|E|}$, whose entries give the Euclidean distance between each pair of vertices connected by an edge. The dihedral angles form a second vector $\theta \in \mathbb{R}^{|E|}$, where each entry records the angle between the normal vectors of the two faces that share an edge. Together, these vectors form the NRIC encoding of a mesh, where the geometry is described using local and intrinsic measurements rather than global vertex coordinates of the conventional mesh representation. The full NRIC vector $z = (l, \theta) \in \mathbb{R}^{2|E|}$ thus lives in a space of dimension twice the number of edges. Since there are more edges than vertices within a mesh, usually around three times as many, we can assume that the same mesh uses a vector twice the size for its NRIC representation than for its conventional representation.

The clearest effect of this encoding is rotation invariance. Edge lengths and dihedral angles are purely intrinsic quantities, so no rigid body transformation applied to the mesh can affect them. Rotating or translating the mesh in 3D space leaves both the edge lengths l and the dihedral angles θ completely unchanged. The NRIC representation therefore assigns a single vector to an entire equivalence class of vertex configurations that differ only by a rigid body motion [20]. This invariance is not imposed by data augmentation or by architectural choices in a downstream model. It follows directly from the representation itself, so any learning algorithm working on NRIC-encoded data has access to it without extra effort.

A second important property is that deformations expressed in NRIC tend to be locally sparse in a way that vertex coordinates are not. Sassen et al. illustrate this with the example of a human character lifting their arm. When this motion is tracked through vertex positions, the induced change propagates across all vertices of the entire arm. When the same motion is expressed in NRIC, the change is concentrated at the shoulder region where the physical work is actually done, while edge lengths and dihedral angles elsewhere remain largely constant [20]. This locality is a key reason to expect NRIC to provide a more natural and compact medium for machine learning on mesh sequences.

A third important property is that NRIC directly encodes physically meaningful geometric relationships. Edge lengths capture local scale and stretching of the surface, while dihedral angles capture local bending and curvature. These are exactly the qualities that physical deformation cares about and which geometric deep learning methods [14, 21, 8] typically spend considerable effort computing from raw vertex data. In NRIC, they are the primary data.

Not every vector $(l, \theta) \in \mathbb{R}^{2|E|}$ can be assigned to a valid mesh embedding in 3D space. For a vector of edge lengths and dihedral angles to correspond to a plausible mesh once reconstructed in 3D space, it has to satisfy the triangle inequalities and a set of discrete integrability conditions more thoroughly described by Sassen et al [20]. These conditions stem from the requirement that local face frames defined by edge lengths and dihedral angles close consistently when traversed around every interior vertex of the mesh [20]. The set of all valid NRIC vectors forms a submanifold defined implicitly by these conditions. Points on this manifold correspond to valid meshes, while points outside it do not directly correspond to any consistent embedding, though approximate vertex positions can still be recovered. The GOAST library [10] provides the tools necessary to perform this reconstruction, as well as the initial transformation from vertex coordinates to NRIC. This reconstruction step is a practical cost of working with this representation and will be discussed in further detail in the context of our experimental setup.

2.3. Euclidean and Non-Euclidean Spaces

Understanding why conventional mesh representations are difficult to work with in machine learning requires a basic understanding of what distinguishes Euclidean from non-Euclidean spaces, as this distinction underlies most of the challenges discussed throughout the thesis.

A Euclidean space is a space that behaves like the familiar flat geometry of $\mathbb{R}^n$. It has a single global coordinate system as well as notions of distance and angle that hold uniformly across the entire space. More importantly for machine learning, however, the concept of translation invariance, in essence, moving a pattern from one location to any other, does not change its fundamental structure. Images are prime examples of data that lives in a Euclidean space. A pixel grid is a regular, flat, uniformly sampled structure, and a convolutional filter applied at one location of an image behaves in the same way when applied at any other location. This shift-invariance is what allows convolutional neural networks to learn features that generalize across spatial positions, and it is the primary reason for success in deep learning and image analysis.

A space that does not share these properties is a non-Euclidean space. The surface of a sphere is a simple example, since there is no global coordinate system that can be spread across it without distortion. Distances and angles change as one moves closer to or further away from the equator or the poles, so there is no consistency in translating a pattern from one location to another. We can describe surfaces like this as manifolds, spaces that are locally similar to a flat Euclidean patch but globally are not. More relevant to our setting, a triangular mesh is a manifold structure of this kind. Its vertices are connected with irregular, varying connectivity, and a neighbourhood around one vertex may look completely different from a neighbourhood around another. Shifting a filter from one vertex to another does not preserve the same geometrical relationship, and as such, convolutional operations cannot be

applied to meshes without significant modification.

Vertex coordinates place data in an Euclidean space ($\mathbb{R}^{3n}$ for a mesh with n vertices), but this does not make the data Euclidean in itself. The coordinates are Euclidean numbers, but the relationship between the vertices is not, as it is governed by mesh connectivity, which is irregular and does not follow a regular Euclidean grid. As such, two geometrically identical meshes that are oriented differently in space will have completely different coordinate vectors. NRIC avoids this issue by working directly with the geometry of the mesh. Because edge lengths and dihedral angles are defined by the mesh surface itself rather than by its embedding in 3D space, they are both nonlinear and rotation-invariant. While the NRIC representation is not a part of a standard Euclidean space either, as not all vectors in $\mathbb{R}^{2|E|}$ correspond to valid meshes, the valid configurations form a well-defined space which is directly tied to the geometry of the mesh rather than the space it inhabits.

2.4. Latent Spaces

A latent space is a low-dimensional continuous space learned by a neural network in which each point represents a compressed encoding of an input. When a model is trained to encode mesh data into latent space and decode it back, similar meshes tend to cluster near each other, and smooth paths through space tend to correspond to plausible interpolations between the shapes at their endpoints.

The dimensionality of the latent space is generally far smaller than the dimensionality of the input, meaning that the model is forced to discard redundant or irrelevant variation in its features and retain only the most meaningful aspects of the data. This makes latent spaces useful not only for compression but also for synthesis, as new points sampled from the space can be decoded into plausible outputs that were not present in the training data. The quality and structure of a latent space depend heavily on the representation the model is trained on, which is one of the central questions this thesis seeks to answer.

Related Work

This chapter surveys the work most relevant to our thesis, organized around three themes: the challenge of applying machine learning directly to conventional mesh representations, rotation-invariant and intrinsic representation for deformation and learning, and latent space methods for 3D mesh synthesis and interpolation.

3.1. Machine Learning on Conventional Mesh Representations

Applying deep learning to 3D shapes represented as vertices and faces is a fundamentally non-trivial problem. As Bronstein et al. observe, the success of deep learning on images comes from exploiting statistical properties such as stationarity and locality that are native to regular Euclidean grids [2]. As previously mentioned, meshes are discrete manifolds which do not follow a regular Euclidean grid, meaning that they possess no global parametrization, no shift-invariance, and have irregular connectivity that varies from one vertex to another. These properties indicate that standard convolutional operations cannot be directly applied to mesh data. Bronstein et al. provide a comprehensive survey of geometric deep learning methods that attempt to address this, broadly categorizing them into spectral approaches, which define convolution through the eigenfunctions of the mesh Laplacian, and spatial approaches, which define convolution through local intrinsic patch operators [2]. Both approaches require considerable additional computation on top of the base vertex and face representation before any learning can begin, whether that means computing the graph Laplacian, constructing local coordinate systems, or defining curvature-based descriptors. This overhead is one of the primary motivations for our investigation into whether a change of representation can reduce the burden placed on the learning model itself.

MeshCNN [8] offers a particularly instructive example of what is required to make convolution work natively on meshes. Rather than operating on vertices, MeshCNN performs convolution on mesh edges using the four incident neighbouring edges of each edge as the convolutional neighbourhood, and defines a symmetric aggregation operation to achieve invariance to rotation, translation, and scale. The input features are not raw vertex coordinates but a five-dimensional vector containing the dihedral angle, the two inner angles, two edge-length ratios, and more. This design choice is motivated by the weaknesses of absolute Cartesian coordinates as input features. When MeshCNN is trained using raw vertex positions instead of relative geometric features, generalization degrades significantly under non-uniform scaling, while the relative features remain robust [8]. MeshCNN demonstrates that the choice of geometric quantities to be fed to a learning model has a larger effect on performance than architectural complexity alone. This further motivates our comparison between raw vertex coordinates and NRIC as learning inputs.

3.2. Rotation-Invariant and Intrinsic Representations

Researchers have studied rotation-invariant and intrinsic representations as alternatives to vertex coordinates, driven by the same limitations that motivate our work.

Tan et al. propose a variational autoencoder framework for performing 3D mesh models that replaces raw vertex coordinates with the Rotation Invariant Mesh Difference (RIMD) feature as its input

representation [22]. RIMD encodes deformation using local rotation differences between adjacent vertices, a per-vertex scaling, and a shear component, thus becoming invariant to global rotation and translation. Tan et al. compare it against vertex-based inputs across several datasets and report that training on RIMD features lowers per-vertex reconstruction error by 59% to 94% relative to raw 3D coordinates and by 27% to 72% relative to globally aligned coordinates. The benefits are not limited to reconstruction. The RIMD-based latent space also supports shape generation, interpolation, and low-dimensional embedding, and on these tasks it performs well beyond what raw vertex coordinates allow. This work is the most direct empirical precedent for our own investigation: it shows that switching to a rotation-invariant representation can yield substantial improvements in learning outcomes, and it establishes RIMD as a competing alternative to NRIC.

Winkler et al. address the problem of interpolating between mesh poses directly, without any machine learning component, and reach conclusions that are closely related to our theoretical motivation [24]. They observe that linear interpolation of vertex positions between two mesh configurations is known to fail as it introduces volume shrinkage, self-intersections, and other artefacts. That is because straight paths in vertex space do not correspond to geometrically plausible deformations. Their solution is to instead linearly blend edge lengths and dihedral angles of the two configurations and then reconstruct vertex positions via a multi-scale shape matching algorithm. This approach produces more physically plausible intermediate shapes because deformation paths in the space of edge lengths and dihedral angles are more linear than their counterparts in vertex space. This observation from a classical geometry processing perspective provides important theoretical grounding for our hypothesis that NRIC may be a more natural medium for machine learning on meshes.

Finally, the NRIC framework itself, introduced by Sassen et al. [20], and described in more detail in the background chapter, forms the technical foundation of our work. Our experimental pipeline relies directly on the GOAST library [10], which implements the forward transformation and the reconstruction algorithms necessary to traverse to and from NRIC and conventional mesh representations.

3.3. Latent Space Learning for Mesh Synthesis

The use of low-dimensional latent spaces to represent and synthesize 3D mesh geometry has become an active area of research, and the quality of these spaces depends significantly on what the learning model is trained on.

As discussed in Section 3.2, Tan et al. show that a variational autoencoder trained on RIMD features learns a well-structured probabilistic latent space that supports shape generation, interpolation, and embedding [22]. Their framework is particularly relevant to our work for a specific reason. It compares the effects of different input representations on the quality of the resulting latent space, the same kind of comparison we make between vertex coordinates and NRIC. For that reason, it serves as a useful methodological template for our study.

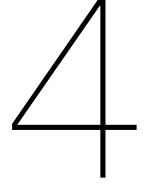
Zhou et al. extended this approach with a fully convolutional mesh autoencoder that uses efficient spatially varying kernels adapted to the irregular structure of triangle meshes [25]. Their architecture learns locally controlled latent representations of deformable surfaces and produces high-quality intermediate shapes via latent space interpolation. The main technical contribution is a kernel design that adapts to the local geometry of the mesh rather than relying on a fixed global parametrization, thus allowing the model to capture fine-grained local deformations that global methods tend to smooth over. Their results demonstrate that with sufficiently expressive architecture, generative models can produce diverse and plausible outputs.

Eisenberger et al. present NeuroMorph, an unsupervised method that learns shape interpolation and surface correspondence at the same time, and it does so without requiring any ground-truth correspondence labels [5]. Given two input meshes, NeuroMorph produces a continuous deformation sequence connecting them while also establishing point-to-point correspondences between the shapes. The approach is relevant because it tackles the joint problem of interpolation and correspondence in a single framework, thus demonstrating that latent space methods can generalize to geometrically diverse inputs, beyond a single object category. Although these results are interesting, the scope of our work is not extensive enough to include experiments for this effect.

Hartwig et al. take a different approach and develop tools for geodesic calculus on the latent manifolds of trained autoencoders [9]. Rather than treating latent space as a flat Euclidean space in which linear interpolation is deforming, they propose to approximate the latent manifold as an implicit subman-

ifold of the ambient latent space and compute geodesic paths on it using discrete Riemannian calculus. Their framework is independent of the underlying autoencoder architecture and can be applied to any training model, making it a potential tool for analyzing and comparing the geometric structure of latent spaces trained on different input representations, including the vertex-based and NRIC-based spaces we construct in our experiments.

Sassen et al. address the related problem of constructing low-dimensional nonlinear shape spaces directly from mesh data through Sparse Principal Geodesic Analysis on the Riemannian manifold of discrete shells [19]. Their method produces a set of modes that represent largely decoupled and spatially localized deformations, such as the bending of individual joints in an articulated figure, and uses these modes to construct a compact nonlinear subspace for interactive deformation. This is relevant both as a non-learning approach to mesh dimensionality reduction and as a perspective for understanding what structural properties a well-organized shape space should exhibit, letting us know what we should look for when comparing NRIC-based and vertex-based latent spaces in our experiments.



Methodology

This chapter describes the methods used to investigate the central research question of the thesis: whether NRIC is a better medium for machine learning on 3D meshes than conventional vertex coordinates. We describe each component in the order in which it was developed, explaining the motivation behind design choices and the challenges that required additional engineering, particularly those that arose when adapting the pipeline to work with NRIC for sequence compression and later the pipeline for vertex coordinates for the variational autoencoder. Section 4.1 covers the mesh compression task, while Section 4.2 covers latent space learning and mesh synthesis.

4.1. Mesh sequence compression via overfitting

The compression task we consider is compression through overfitting. Rather than training a general-purpose model on a large dataset, a small neural network is trained exclusively on a single mesh sequence, with the goal of encoding the entirety of the animation as accurately as possible within the model’s weights. At evaluation time, the trained network is queried at each timestep to reconstruct the corresponding frame.

This setting works well as a controlled test of how learnable each coordinate system is for a neural network. Since the network is never asked to generalize beyond the training data, the task comes down to a single question: how efficiently, and above all, how accurately, can a given coordinate system be memorized by a compact model? If NRIC deformations really are more linearly structured, as we hypothesize, then a smaller network should reach a lower reconstruction error in NRIC space than in vertex space. If instead NRIC brings complications that outweigh any structural benefit, we should see the opposite, either a higher reconstruction error or a need for more model capacity.

4.1.1. Network Architecture

The network used in both the vertex and NRIC experiments is a simple multilayer perceptron (MLP), which we refer to as *SimpleNet*. The network takes a scalar time value $t \in [0, 1]$ as input and outputs a prediction of the mesh state at that time, either as a matrix of vertex positions $\hat{V} \in \mathbb{R}^{|V| \times 3}$ or as a matrix of NRIC features, $\hat{Z} \in \mathbb{R}^{|E| \times 2}$, depending on the experiment. The architecture consists of a linear input projection, a fixed number of hidden layers, each containing a linear transformation and a ReLU activation, and a final output layer. The architecture, available in Figure 4.1, is kept identical across both representations, so that any difference in performance can be attributed to the coordinate system, rather than the model capacity.

4.1.2. Fourier Time Encoding

A scalar time value $t \in [0, 1]$ is a poor input for a standard MLP. Neural networks with ReLU activations have a natural bias towards learning low-frequency functions of their inputs, a property studied by Rahaman et al. [17]. However, a mesh sequence may contain fine temporal deformations that require networks to capture high-frequency variations in the output. Providing only the raw scalar t limits the network’s ability to do so.

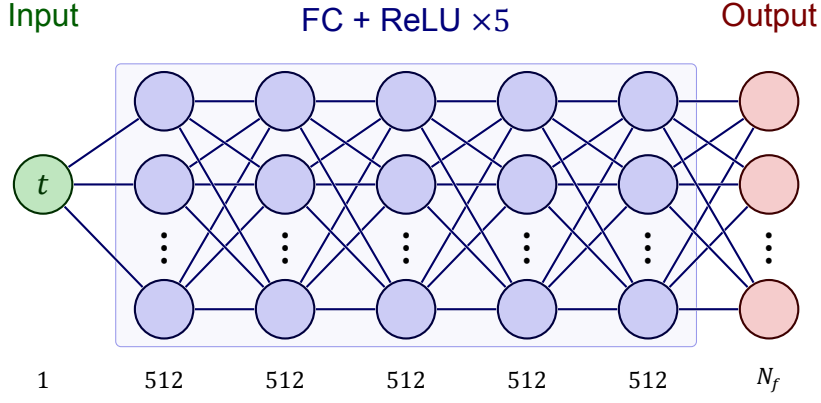


Figure 4.1: Initial *SimpleNet* architecture, where N_f represents the number of output features: $N_f = |V| \times 3$ for raw vertex positions, and $N_f = |E| \times 2$ for NRIC features. This MLP architecture was the first iteration used to evaluate which medium is better for compressing mesh sequences.

To deal with this, we use positional Fourier encoding, which turns the scalar input into a higher-dimensional feature vector before it enters the MLP. We first define a set of L fixed frequency bands

$$\omega_k = 2^k \cdot \pi, \quad k = 0, 1, \dots, L-1,$$

and then embed t as

$$\gamma(t) = [\sin(\omega_0 t), \cos(\omega_0 t), \sin(\omega_1 t), \cos(\omega_1 t), \dots, \sin(\omega_{L-1} t), \cos(\omega_{L-1} t)] \in \mathbb{R}^{2L}.$$

The raw scalar is now replaced by a $2L$ -dimensional vector that presents the MLP with frequencies spanning several orders of magnitude, which lets it capture fine-grained temporal structure. In our experiments, we set $L = 8$ frequencies, giving the MLP a 16-dimensional input. This means that our Fourier encoding can faithfully represent up to ~ 128 distinct temporal positions. While the sequences we test on, such as DFAUST [1], can contain between 200 and 600 frames, the network is not expected to encode each frame independently, but rather to learn a smooth mapping from time to motion, thus operating at the level of motion variation rather than individual time frames. Increasing L beyond 8 would add unnecessary capacity without meaningfully improving reconstruction, something our experiments also suggest. Figure 4.2 shows the final architecture of *SimpleNet* with the Fourier encoding introduced.

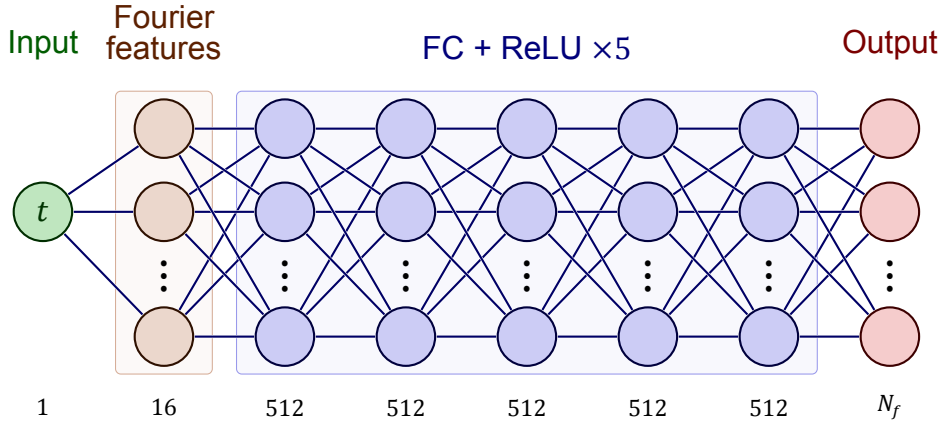


Figure 4.2: Final *SimpleNet* architecture which also includes Fourier encoding of time input. N_f represents the number of output features: $N_f = |V| \times 3$ for raw vertex positions, and $N_f = |E| \times 2$ for NRIC features. This MLP architecture was used in all experiments that evaluate which medium is better for compressing mesh sequences.

4.1.3. Vertex Coordinate Approach

For the vertex coordinate baseline, training is simple. The model is optimized to minimize the mean squared error between the predicted and ground-truth vertex positions, taken across all frames and

all vertices at once. Training on the full dataset rather than on randomly sampled batches speeds up convergence. This, however, is not workable for sequences with many frames, such as the crumpling paper sequence [6], so in those cases we fall back to batch training. We let $V_i \in \mathbb{R}^{|V| \times 3}$ denote the ground-truth vertex positions for frame i and $\hat{V}_i = f_\theta(t_i)$ the model prediction. The training objective then becomes

$$L = \frac{1}{3NV} \sum_{i=1}^N \|V_i - \hat{V}_i\|^2,$$

where N is the number of frames and V is the number of vertices. The model is trained using the Adam optimizer [12], with a learning rate of 5×10^{-4} for 20,000 iterations, with the full set of sequence frames as its training set. We intend to only assess the impact of raw vertex coordinates and, as such, no preprocessing steps are required. A visual representation of the entire pipeline can be seen in Figure 4.3.

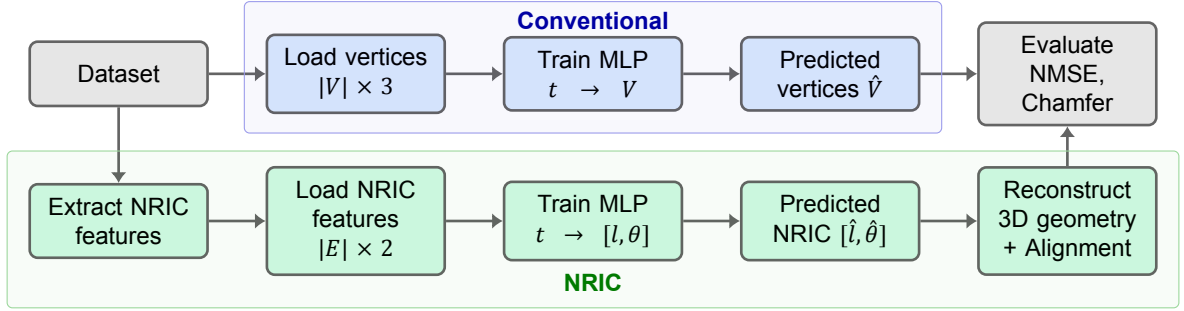


Figure 4.3: Shared pipeline for conventional and NRIC approaches. Gray nodes represent common steps, blue nodes represent conventional steps, and green nodes represent NRIC steps.

4.1.4. NRIC Approach

Adapting the same framework to NRIC required several additions that were not necessary for the vertex coordinate baseline. Those arose from two sources: the different statistical properties of edge lengths and dihedral angles as learning targets and the need to convert predicted NRIC features back into valid vertex positions for evaluation. The complete NRIC pipeline can also be seen in Figure 4.3 alongside the conventional one.

Feature Normalization. Edge lengths and dihedral angles operate on fundamentally different numerical scales. Edge lengths are determined by the physical size of the mesh and are strictly positive, while dihedral angles are bounded to $[-\pi, \pi]$ and are often much smaller in magnitude than the edge lengths for most mesh sequences. When both are passed to an MLP as regression targets, a single loss forces the optimizer to fit both scales at the same time. In practice, the larger-scale features dominate the gradient signal, and the network struggles to learn the smaller-scale dihedral angles accurately. We deal with this by standardizing both sets of features before training, normalizing each feature vector to zero mean and unit standard deviation over all frames of the sequence. The normalization statistics are stored and then used to unnormalize the predictions at evaluation time.

Loss Balancing. After normalization, the edge lengths and dihedral angles sit on comparable numerical scales, but they still do not contribute equally to the final reconstruction error. The two channels describe geometrically distinct quantities, and an error of a given magnitude in the edge lengths affects the reconstructed mesh more than an error of the same magnitude in dihedral angles. This raised the question of whether the loss should treat the two channels differently rather than weigh their errors carefully.

We tried two weighing schemes before deciding against both. The first applied a fixed scalar coefficient λ to scale the dihedral angle loss against the edge loss. It managed to reduce the dominance of one term over the other, but the value of λ that worked best changed from sequence to sequence and had to be set by hand for each dataset, so it never held up as a general solution. A second attempt

learned the weights during training rather than fixing them, relying on per-edge uncertainty parameters in the style of Kendall et al. [11]. Although this meant that manual tuning could be avoided, it brought along problems of its own. The extra parameters made the optimization more complex and slowed training down, and, to our surprise, even reduced the accuracy of the model. We therefore returned to a plain loss that equally weights the edge length and dihedral angle errors, keeping the training procedure simple and matching the accuracy of both weighted variants. This loss is identical to the one in Section 4.1.3, but compares predicted NRIC frames to the ground-truth NRIC features instead of the 3D vertices. Then the loss is

$$L = \frac{1}{2NE} \sum_{i=1}^N \left(\|l_i - \hat{l}_i\|^2 + \|\theta_i - \hat{\theta}_i\|^2 \right),$$

where N is the number of frames, E is the number of edges, l_i and $\hat{l}_i$ are the edge lengths predictions and ground-truths for frame i , and θ_i and $\hat{\theta}_i$ are the dihedral angle predictions and ground-truths for frame i .

Reconstruction. In contrast to training our model using raw vertex coordinates, training it on NRIC also produces a prediction in NRIC space. Therefore, at evaluation time, the unnormalized predicted edge lengths $\hat{l}$ and dihedral angles $\hat{\theta}$ are passed to the GOAST library to be reconstructed into a set of vertex positions in 3D space. For this task, there are two possible approaches: First, direct reconstruction, which immediately begins reconstructing the geometry of the mesh from the orientation of a set starting face (usually the face at index 0) regardless of the way vertex positions are arranged during the reconstruction, leading to possible artefacts. The second option is least-squares reconstruction, where a solver searches for a configuration that best fits the geometric shape of the mesh given the predicted edge lengths and dihedral angles. Its advantage is that the resulting meshes are more geometrically coherent, at the cost of a more expensive reconstruction process.

Postprocessing. As mentioned, reconstruction from NRIC begins with a set face in order to provide the final mesh with global alignment similar, or preferably, identical to the ground truth for the same frame. Although the 3D meshes reconstructed from NRIC are not completely free-floating in 3D space, in practice, we observed that small errors in the predicted edge lengths and dihedral angles accumulate during reconstruction and manifest a subtle frame-to-frame discrepancy when the frames are played back as a sequence. Although each frame is reconstructed independently, the residual misalignment between consecutive frames is perceptible in the resulting animation and, more importantly, in the evaluation metrics. Since this is a product of NRIC’s rotation-invariant nature, and not a product of training, we intend to mitigate this in postprocessing, hopefully creating a better-suited environment for a fair comparison between the two mediums. The Kabsch algorithm computes the rotation and translation that minimize the RMSE between two vertex sets, using a singular value decomposition of the cross-covariance matrix. Besides improving visual coherence, this alignment also lowers the measured reconstruction error by removing any rigid-body component that does not come from the accuracy of the network’s predictions.

4.2. Variational Mesh Encoding

The compression task trains a dedicated network on a single sequence in order to memorize it. Latent space learning asks a fundamentally different question. Can a single model, trained across a whole collection of meshes, learn a structured low-dimensional space in which every mesh is a point, nearby points correspond to geometrically similar meshes, and new points can be decoded into plausible new geometries? Moving from per-sequence memorization to shared generalization is the main challenge of this second task. Here, our goal is not compression fidelity but the organization of geometry into a continuous space that can be explored, interpolated, and sampled from. We use a Variational Autoencoder (VAE) for this purpose, training both a conventional vertex coordinate and an NRIC variant under the same architecture and training procedure, after which we compare the structure and quality of the resulting latent spaces.

For NRIC, these goals carry an additional expectation rooted in the properties of the representation itself. Because edge lengths and dihedral angles are invariant to rigid body transformations, the

NRIC feature vector contains no information about where the mesh is positioned or how it is oriented in 3D space, only about its intrinsic geometry. A latent space learned from these features should therefore be organized purely by shape, with no capacity wasted on encoding orientation variation that is geometrically meaningless. We expect this to produce a more tightly structured space compared to the conventional approach, where the encoder must isolate the true pose variation from the rigid-body component in every mesh. Beyond structure, the individual dimension of an NRIC-based latent space may also encode motion in a more natural and geometrically interpretable way. Because deformations in NRIC space are locally sparse and concentrated at the sites of physical change, we expect the latent encoding of a deformation to more directly reflect where and how the geometry is actually moving, rather than spreading that signal diffusely across all vertex coordinates simultaneously. Whether these expectations are borne out in practice is one of the central questions the experiments in Section 5.3 aim to answer.

4.2.1. Network Architecture

The VAE is made up of an encoder and a decoder, both implemented as MLPs. The encoder takes a flattened feature vector (vertex positions $\mathbb{R}^{|V| \times 3}$ or concatenated NRIC features $\mathbb{R}^{|E| \times 2}$) and passes it through three consecutive hidden layers of different sizes with ReLU activations. It then splits into two parallel linear projections that produce the mean vector $\mu \in \mathbb{R}^d$ and the log-variance vector $\log \sigma^2 \in \mathbb{R}^d$ of the approximate posterior over the latent space, where d is the latent dimension. The decoder maps the latent vector z back to the output feature space through a symmetric MLP that reverses this layer structure, with hidden layers of the same reversed sizes, a final linear projection, and a Tanh activation. We set the latent dimension d to 24 or 48 for the smaller datasets and to 64 for the largest, in line with how much variability each dataset contains.

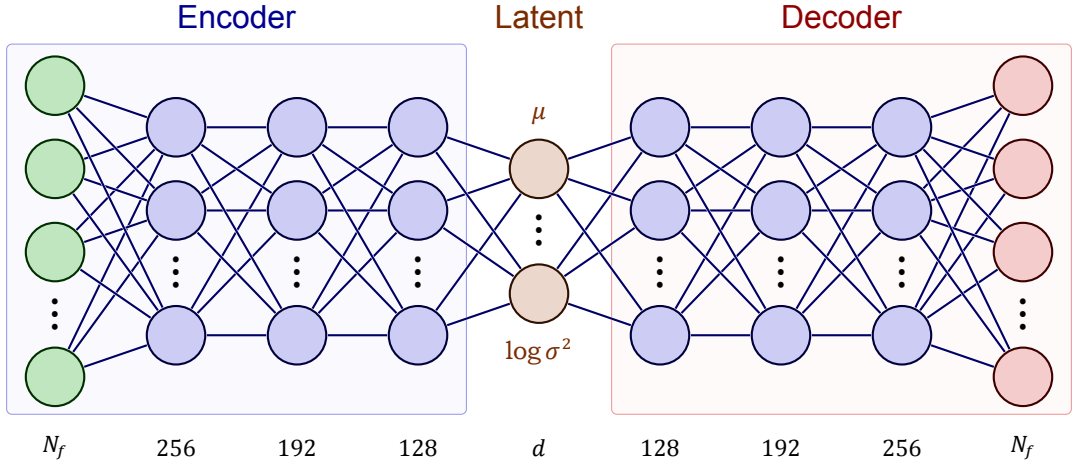


Figure 4.4: Variational Autoencoder (VAE) architecture. N_f represents the number of output features: $N_f = |V| \times 3$ for raw vertex positions, and $N_f = |E| \times 2$ for NRIC features. d is the size of the latent dimension, usually 24 or 48 for smaller datasets, and 64 for larger datasets. The sizes for the hidden layers depend on the dataset. In this figure, they are configured for the MANO right hand dataset.

Unlike the compression network, which used five hidden layers to memorize a single sequence, the encoder here only uses three. The shallower architecture is deliberate. In the compression setting, the network must store an entire animation sequence in its weights, and depth is necessary to capture fine temporal variation. In the VAE setting, the encoder does not need to memorize, it needs to extract the most relevant geometric variation across a collection into a compact code. In this setting, the main compression constraint is the latent dimension itself, not the depth of the network, and a shallower architecture lowers the risk of overfitting to individual meshes before the bottleneck has had the chance to force a meaningful structure. A latent sample z is drawn from the encoder’s predicted distribution with the reparametrization trick,

$$z = \mu + \sigma \odot \epsilon, \epsilon \sim N(0, 1)$$

where $\sigma = \exp(0.5 \times \log \sigma^2)$ and $\odot$ denotes element-wise multiplication. This formulation rewrites the sampling step as a deterministic transformation of a fixed noise source, allowing gradients to flow

through z and back into μ and $\log \sigma^2$ during training. Without this, the stochastic sampling would block gradient propagation, and the encoder could not be trained by backpropagation. A diagram of the entire architecture is shown in Figure 4.4.

The architecture is kept identical across the conventional and NRIC variants. The only difference is the input and output dimensionality, which scales with the number of vertices in the conventional approach and twice the number of edges in the NRIC approach. As in the compression experiments, this allows any difference in latent space quality to be attributed to the coordinate system rather than the model.

This architecture is inspired by the variational autoencoder framework of Tan et al. [22], adapted here to operate on flattened feature vectors rather than the RIMD based input of the original work. Where Tan et al. design their encoder and decoder around the mesh connectivity to exploit local geometric structure, our network treats the input as a flat vector of either vertex coordinates or NRIC features, keeping the architecture representation agnostic and comparable across the two coordinate systems.

4.2.2. Preprocessing

The two coordinate representations enter the training pipeline with different statistical properties and different sources of unwanted variation, and preprocessing is applied to address these separately before any feature is passed to the network. Not all steps are shared between the two approaches, and where they differ, the motivation is specific to the nature of each representation.

Wrist Alignment for MANO dataset. For the right hand models from the MANO dataset [18], all meshes are aligned to a canonical orientation before training. While the MANO dataset provides a consistent mesh topology across all poses, it contains no constraint on the global orientation of said poses, meaning that two meshes representing the same hand pose may differ only by a rotation in 3D space, yet their vertex coordinate vectors would be entirely different. For the vertex coordinate approach, this orientation variation is something the encoder must account for, consuming latent capacity on information that is geometrically meaningless. Wrist alignment removes this degree of freedom.

The procedure uses 16 vertices along the wrist boundary, the same indices across all hand meshes, as a reference anchor. Their centroid is computed and subtracted from every vertex position, which moves the mesh so that the wrist center sits at the origin. The normal to the plane formed by these wrist vertices is then estimated with an SVD, where the right singular vector that belongs to the smallest singular value gives the direction most orthogonal to the spread of the wrist boundary and is taken as the wrist plane normal. Its sign is resolved by checking whether it points towards the bulk of the hand, and it is flipped if it does not. A rotation that aligns this normal with the z-axis is then applied to all vertex positions. The result places every mesh in a consistent frame, with the wrist roughly centered at the origin and the hand extending in the positive z-direction.

This alignment is applied to both representations at data loading time, but only affects the training of one. For the conventional vertex coordinate approach, it directly reduces the variation the encoder must explain, being that meshes in the same pose now have nearly identical vertex coordinate vectors regardless of their original orientation. For NRIC, edge lengths and dihedral angles are already invariant to rigid body transformations, so the alignment has no effect on the features used during training. It is nonetheless applied to the stored vertex positions used at evaluation time, ensuring that reconstructed geometries are comparable to ground truth in the same consistent frame.

Feature Normalization. For NRIC, each edge length and dihedral angle is normalized per feature across the full dataset with min-max scaling,

$$\tilde{x}_i = 2a \times \frac{x_i - \min_i}{\max_i - \min_i + \epsilon} - a,$$

with $a = 0.9$, which maps all features into $[-0.9, 0.9]$. We choose this range to match the bounded output of the Tanh decoder activation, so that the decoder can cover the full dynamic range of the data without saturating. The margin of 0.1 on each side keeps values near the extremes of the training distribution from being clipped. The normalization statistics are computed once from the full dataset before training and kept on hand, so that the predictions can be returned to their original scale at

evaluation time. For conventional vertex coordinates, we apply no explicit normalization beyond the wrist alignment. That alignment already centers the mesh at the origin and keeps vertex positions within a numerically consistent range across the dataset, which is enough for the raw vertex approach, since it uses no bounded output activation.

4.2.3. Training Objective

The VAE is trained to maximize the Evidence Lower Bound (ELBO), a lower bound on the log-likelihood of the data under the generative model. The ELBO has two terms. The reconstruction term measures how faithfully the decoder reproduces the input from the latent code, and the KL divergence term measures how far the encoder’s approximate posterior departs from the prior. Maximizing the ELBO pushes for accurate reconstruction and, at the same time, for a well-organized latent space in which the posterior of each input stays close to a standard normal.

The reconstruction loss is the mean squared error between the input feature vector x and its reconstruction $\hat{x}$,

$$L_{\text{recon}} = \frac{1}{|x|} \sum_i (\hat{x}_i - x_i)^2,$$

where the sum runs over all elements of the flattened input. The KL divergence from the approximate posterior $q(z|x) = N(\mu, \text{diag}(\sigma^2))$ to the standard normal prior $p(z) = N(0, 1)$ evaluates analytically per latent dimension j as

$$KL_j = -0.5 \times (1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2),$$

averaged over the batch. The full training objective is

$$L = L_{\text{recon}} + w_{KL} \times \frac{1}{d} \sum_j KL'_j,$$

where KL'_j is the per-dimension KL after applying the free bits described in Section 4.2.4, d is the size of the latent dimension, and w_{KL} is the selected KL weight. The model is trained using Adam with a learning rate of 5×10^{-4} , with mini-batches of 64 meshes for the smaller datasets and 128 meshes for the largest. Unlike the compression experiments, which used full-batch training where practical, the VAE requires mini-batching because the dataset is substantially larger and the gradient signal from the reconstruction loss is already informative per batch without seeing the entire dataset each step.

4.2.4. Free Bits

A known failure mode of VAEs is posterior collapse, where the encoder learns to produce posteriors that are identical to the prior of $N(0, 1)$ for every input, and the decoder learns to ignore the latent code entirely and produce only the dataset average. This occurs when the KL term is penalized too strongly relative to the reconstruction term, causing the optimizer to set KL to zero by collapsing all encoder output towards the prior. The result in the latent space is that most of the dimensions carry no information, that their per-dimension KL is near zero, while a small number of dimensions absorb all the variation in the data, producing highly uneven and poorly structured space.

Free bits handles this by leaving any dimension unpenalized when its KL contribution falls below a threshold λ ,

$$KL'_j = \max(KL_j, \lambda).$$

This floor keeps dimensions at or below the threshold out of the KL loss and lets them build up useful structure on their own, without being pushed back toward the prior. Only dimensions that already exceed the threshold, those carrying meaningful information, are penalized. The resulting effect is that the encoder is encouraged to spread variation across at least as many dimensions as the threshold allows before any dimensions collapse. The threshold λ is tuned per dataset and representation, with the conventional approach generally requiring a higher value than NRIC. This points at how much harder it is to organize raw vertex coordinates into a structured posterior. Without a higher threshold, more dimensions tend to collapse early in the training, so a larger free-bit floor is needed to keep the latent space well populated throughout.

5

Experimental Results

This chapter presents the experiments carried out to answer our main research question and its sub-questions. It opens with a description of our Python bindings setup, which gives our preferred Python codebase access to the GOAST functions related to NRIC, and then moves into a section for each set of implementation results, one for sequence compression and one for latent space synthesis with the variational autoencoder. Each of these sections begins with the experimental setups specific to that task, followed by an interpretation of the experiments, and closes with a discussion that ties the findings back to the questions we want to answer.

5.1. GOAST Python bindings

GOAST [10] is a C++ library implementing the NRIC forward map, mesh reconstruction operations, and more. To integrate the necessary functions into the Python-based machine learning pipeline we intended to create, a set of Python bindings was developed using pybind11 [23], exposing the library's core functionality without requiring the training code to leave the Python environment. The binding layer converts between GOAST's internal Eigen vector types and NumPy-compatible matrices, which lets NRIC data move straight into PyTorch tensors without extra steps. We chose to use bindings rather than work directly in C++ for the benefit of working in a more familiar environment and rely on CUDA [16] to cut down the runtime of training and evaluation.

Two structured return types are exposed through bindings. `MeshData`, stores vertex positions and face indices as NumPy arrays, while `NRICData` stores the edge lengths and dihedral angles separately. The bindings provide three core functions:

- Function `read_mesh_data` reads an OBJ or PLY file and returns the vertex and face data.
- Function `read_nric_data` reads a mesh, applies the NRIC map internally, and returns the resulting edge lengths and dihedral angles.
- Functions `reconstruct_nric_geom` and `reconstruct_nric_geom_least_squares` both take a reference mesh file and an `NRICData` object and return the reconstructed vertex positions.

Both reconstruction operators need a reference mesh to fix the topology and to anchor the global position and orientation, since NRIC encodes only local geometry and leaves translation and rotation undetermined. The anchor in each case is vertex zero together with its first adjacent face on the reference mesh. From there, direct reconstruction propagates the local frames outward in a deterministic order. Least squares reconstruction takes a different approach, solving a global quadratic energy that combines the NRIC integrability conditions with a bending regularizer through Gauss-Newton, with the first frame of a sequence serving as a warm start. One finding here ran against our expectations. In our tests, warming up from the previously reconstructed frame proved to be less effective than warming up from a ground-truth frame such as the starting frame. One reason for this occurrence could be that the solver falls into a local minima produced by the predicted and possibly erroneous previous frame, making it harder for new reconstructions to escape it.

Direct versus Least Squares Reconstruction. To evaluate which reconstruction method is more appropriate for the following experiments, both reconstructions are compared on three representative sequences: a synthetic low-resolution mesh (600 vertices), a Dynamic FAUST [1] human motion sequence (6,890 vertices), and a strip of the paper crumpling sequence (19,650 vertices). Reconstruction is evaluated on predicted sequences rather than ground-truth sequences, since the comparison is only meaningful in the presence of real training error. The goal is to understand how each reconstruction method interacts with the imperfections naturally produced by the network and, therefore, both receive identical predictions as input, isolating the reconstruction step as the sole variable. Results are displayed in Table 5.1.

Sequence [#frames]	Direct			Least squares		
	Chamfer	NMSE	NRIC Loss	Chamfer	NMSE	NRIC Loss
Low Poly [48]	1.84×10^{-3}	3.36×10^{-7}	1.44×10^{-6}	5.62×10^{-4}	3.00×10^{-8}	1.44×10^{-6}
CW-02 [216]	1.33×10^{-2}	1.23×10^{-5}	1.86×10^{-6}	6.59×10^{-2}	1.55×10^{-4}	1.86×10^{-6}
Paper-S [107]	8.80×10^{-4}	7.26×10^{-5}	9.10×10^{-8}	6.22×10^{-2}	2.62×10^{-3}	9.10×10^{-8}

Table 5.1: Direct and Least Squares reconstruction comparison. Results of runs over the synthetic low polygon human sequence, DFAUST chicken-wings motion of subject 50002, and a strip of the paper crumpling sequence. The experiments were performed on the same predicted sequences, isolating the difference in error to the reconstruction method. The results display a decrease in overall error when using direct reconstruction for the more complex models in terms of vertex count and sequence length. This experiment was only run once, but its results are consistent with all of the experiments done throughout this thesis.

For the synthetic mesh, least squares reconstruction has lower Chamfer distance and NMSE, suggesting that for low-resolution meshes, where there is a small number of edges to propagate error, the global optimization produces a marginally better layout. However, the improvement is small and approaches the negligible error level to which direct reconstruction is already close.

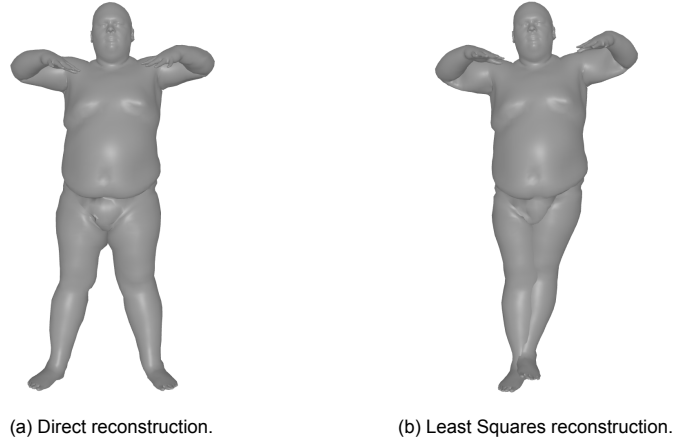


Figure 5.1: Frame 45 of the DFAUST chicken wings animation sequence reconstructed using direct and least squares reconstruction. Both reconstructions use the same NRIC feature vector, and the least squares reconstruction uses frame 0 as a warm start.

The situation reverses for higher-resolution sequences. On the DFAUST sequence, least squares gives a Chamfer distance about five times larger than direct reconstruction, and the paper strip sequence degrades by a comparable amount. In both cases, the NRIC loss stays low, so the predicted features themselves are accurate. The error growth is therefore not a flaw in the least squares method itself, but it comes from the small imperfections in the predicted NRIC features being resolved differently by the two solvers. Direct reconstruction propagates the local frames in a fixed traversal order, so positional error is absorbed into the global placement of each vertex relative to their neighbour. Least squares looks for a globally coherent shape that satisfies the integrability conditions, and on a high-resolution mesh with complex articulations, this can place whole limbs or surface regions in a geometrically consistent but globally displaced position whenever the predicted features are slightly

inconsistent. An example of this behaviour is shown in Figure 5.1

For this reason, direct reconstruction is used throughout all compression experiments. For the variational autoencoder experiments in Chapter 5.3, least squares reconstruction is used instead. When synthesizing novel meshes from latent space, the network has not seen the target geometry and the predicted NRIC features are not constrained to be consistent with any particular sequence. In that setting, producing a geometrically coherent shape, one in which local surface geometry is plausible, is more valuable than pinning global position, thus making the behaviour of least squares reconstruction preferable.

5.2. Mesh Sequence Compression

Mesh sequence compression experiments follow the overfitting procedure described in Chapter 4.1: a single network is trained per sequence, with no generalization to unseen data. The experiments that follow fall into two groups. Some explore different approaches to applying this procedure, testing variations in how the network is trained and how its inputs and outputs are framed. Others are designed to directly answer the research sub-questions proposed for this task.

5.2.1. Experimental Setup

All values reported in our mesh sequence experiments are averaged over 3 independent training runs, with their standard deviations in the complete tables in Appendix A. The variance between runs of the raw vertex coordinate models is small, well below the mean of the experiments, indicating that their results are stable. The results of the NRIC models vary more in terms of 3D reconstruction metrics, which is consistent with the reconstruction step amplifying small differences in the predicted features, but this variance is still not as large as the gap between the two representations, and thus, it does not change the ordering of any of the interpretations done on these results.

The main evaluation metric is the mean Chamfer distance per frame, which measures the average nearest-neighbour distance between the predicted and ground-truth vertex sets. We also report Normalized Mean Squared Error (NMSE), which lets us compare error across datasets of different magnitudes. Chamfer distance is our preferred metric for two reasons. Its nearest-neighbour averaging deals better with isolated vertex outliers than squared error does, and it does not require strict vertex correspondence, which makes it less sensitive to the residual rigid misalignment left over after Kabsch alignment in the NRIC pipeline. Given the scale of the meshes used in our experiments, Chamfer distances below $\approx 1 \times 10^{-3}$ correspond to reconstructions whose errors are generally not visible in rendered animations, while errors above 5×10^{-3} start to produce noticeable geometric artefacts. Similarly, NMSE values around 10^{-8} are effectively negligible, while values above 10^{-4} become noticeable in the reconstructions. For the NRIC experiments, we additionally report the mean NRIC-space MSE, which acts as a diagnostic for measuring how accurately the network predicts the intrinsic feature vector before 3D reconstruction.

Datasets The nine sequences used across all compression experiments fall into three dataset categories, chosen to cover a wide range of mesh sizes, deformation types, and motion complexities.

- **Synthetic Sequences.** The synthetic sequences were generated from a freely available rigged humanoid model [15], to which a repeating flailing arm cycle of approximately 48 frames was applied. Two resolution variants were produced: the original mesh at around 600 vertices (Low Poly) and a remeshed version at 1200 vertices (High Poly). These sequences serve as a controlled baseline, involving smooth and predictable deformation at low vertex counts where both representations (NRIC and vertex coordinates) should be tractable.
- **Dynamic FAUST Sequences (DFAUST).** The human body sequences are drawn from the DFAUST dataset [1], a large-scale collection of registered 3D human body scans in motion. All sequences used share a fixed topology of 6,890 vertices, and our sequences used in our experiments range from approximately 200 to 300 frames. Five sequences were selected with two specific comparisons in mind. The first comparison focuses on four motions performed by subject 50002 (chicken-wings, punching, and running-on-spot), allowing the effect of motion type to be isolated while holding the subject constant. The second comparison consists of the chicken-wing move-

ment from subject 50004, allowing the effect of body shape to be isolated while holding the motion constant.

- **Paper Crumpling Sequence.** The paper crumpling sequences originate from a single large-scale dataset of a full sheet of paper being crumpled, consisting of 154,449 vertices over 106 frames. Directly training on the full mesh is computationally costly at the network sizes used in our experiments, and as such, we decided not to use it in its complete form. Instead, two spatially restricted subsets were extracted: a corner region of the paper at 2,450 vertices (Paper-C) and a strip section at 19,650 vertices (Paper-S). These subsets preserve the near-isometric bending characteristics of the full crumpling sequence while remaining tractable.

To account for the substantial differences in vertex count across the three data categories, the hidden layer width of the network is adjusted accordingly: 256 units per layer for the synthetic sequences, 512 for the DFAUST sequences, and 1024 for the paper sequences. All other architectural and training hyperparameters are kept constant.

5.2.2. Direct Time-to-Feature Mapping

The first experiment is the most direct instantiation of the compression approach: the network takes a normalized scalar timestep $t \in [0, 1]$ as input and is trained to predict the entire coordinate representation of the mesh at that time, either as absolute vertex positions for the conventional model or as absolute edge lengths and dihedral angles for the NRIC model. This is used as a baseline against which subsequent experiments are measured. Table 5.2 shows the results for this experiment, and Figure 5.2 shows a comparison in reconstruction for both representations.

Sequence [#frames]	Raw vertex position		NRIC		
	Chamfer	NMSE	Chamfer	NMSE	NRIC Loss
Low Poly [48]	5.80×10^{-4}	1.85×10^{-8}	1.84×10^{-3}	3.36×10^{-7}	1.44×10^{-6}
High Poly [48]	4.07×10^{-4}	8.96×10^{-9}	9.07×10^{-4}	5.64×10^{-8}	1.07×10^{-7}
CW-02 [216]	2.73×10^{-3}	2.28×10^{-7}	1.33×10^{-2}	1.23×10^{-5}	1.86×10^{-6}
CW-04 [216]	2.46×10^{-3}	2.43×10^{-7}	1.32×10^{-2}	1.00×10^{-5}	1.29×10^{-5}
Punch-02 [303]	2.64×10^{-3}	2.16×10^{-7}	4.69×10^{-2}	1.60×10^{-4}	5.02×10^{-5}
Run-02 [331]	4.07×10^{-3}	3.86×10^{-7}	4.83×10^{-2}	1.33×10^{-4}	1.46×10^{-4}
Paper-C [107]	7.69×10^{-4}	5.12×10^{-6}	5.01×10^{-4}	7.64×10^{-6}	6.06×10^{-6}
Paper-S [107]	2.35×10^{-3}	3.39×10^{-6}	8.80×10^{-4}	7.26×10^{-5}	9.10×10^{-8}
Mean	2.09×10^{-3}	1.09×10^{-6}	1.48×10^{-2}	4.42×10^{-5}	2.56×10^{-5}

Table 5.2: Direct Time-to-Feature Mapping results. Shows results for the low and high polygon variants of our synthetic sequence. CW-02 and CW-04 are the chicken-wings sequences for subjects 50002 and 50004, respectively; Punch-02 and Run-02 are the punching and running-on-spot sequences for subject 50002; and, finally, Paper-C and Paper-S are the corner and strip sequences for paper crumpling. Each sequence is denoted with the number of frames of each sequence, and the better results for each category are denoted in bold. Values are the mean over 3 runs, and the complete table containing the standard deviations can be found in Appendix A.1.

Before discussing individual results, two methodological points bear on how results should be read across dataset categories. First, as previously mentioned, the hidden layer width differs per category as a means to complement the general sizes of the datasets. Hidden Layer widths were chosen based on tests performed for each dataset category, creating a more custom and suitable environment for each of them. Second, the synthetic sequences only span 48 frames, while the DFAUST sequences span 250-300 frames. Shorter sequences are substantially easier to overfit, and the low error on synthetic sequences reflects this.

On the synthetic sequences, both methods stay well within the negligible range for both metrics. These sequences were built to check that NRIC can be compared to vertex coordinates in a meaningful way at small model sizes, and the results confirm it. The two representations perform at similar orders of magnitude, with the conventional one slightly ahead.

The DFAUST sequences are the primary focus of the compression experiments, and most of the engineering effort described in Chapter 4 was directed at making NRIC capable of handling this data. Even so, NRIC performs consistently worse than conventional coordinates across all five sequences, with the gap spanning several orders of magnitude in NMSE. The sequence length appears to play a role in how pronounced this gap is. The chicken wings sequence has around 200 frames, and NRIC NMSE sits around 10^{-5} . The punching and running sequences are substantially longer at around 300 frames, and their NRIC NMSE falls around 10^{-4} , around one order of magnitude worse. Both of the longer sequences involve considerably more whole-body displacement, so sequence length and motion complexity are both confounded here, but the pattern is consistent. Raw vertex coordinates also show some sensitivity to sequence length, but less so than NRIC, only falling one order of magnitude below our synthetic sequences. We assume that part of the difficulty for NRIC on this data is structural, as NRIC represents each mesh with twice as many features as the vertex coordinate representation, the network’s output space is larger for the same mesh, an aspect that we will discuss further later in this paper. The NRIC Loss values, however, indicate that the network also struggles with learning the intrinsic features reasonably well, the drop in 3D quality mostly originating from the reconstruction step rather than from the network’s predictions.

The paper sequences show the clearest advantage for NRIC, which matches our expectations. DFAUST motions involve cyclic patterns, scanned skin surface dynamics, and globally non-rigid body movement. In contrast, the paper crumpling sequence is a direct, non-repeating transition from one geometric state to another, driven almost entirely by surface bending with very little stretching. This is exactly the case that NRIC’s intrinsic encoding of edge lengths and dihedral angles suits best, since the edge lengths stay largely constant throughout the sequence and the deformation signal is concentrated in the dihedral angles. The NRIC loss for the paper sequences is the lowest in the table, which confirms that the intrinsic features are fitted accurately and that the reconstruction step adds little additional error. Raw vertices, on the other hand, have to express the same bending as correlated displacements across many vertices, which is a harder regression target that leads to visibly higher error.

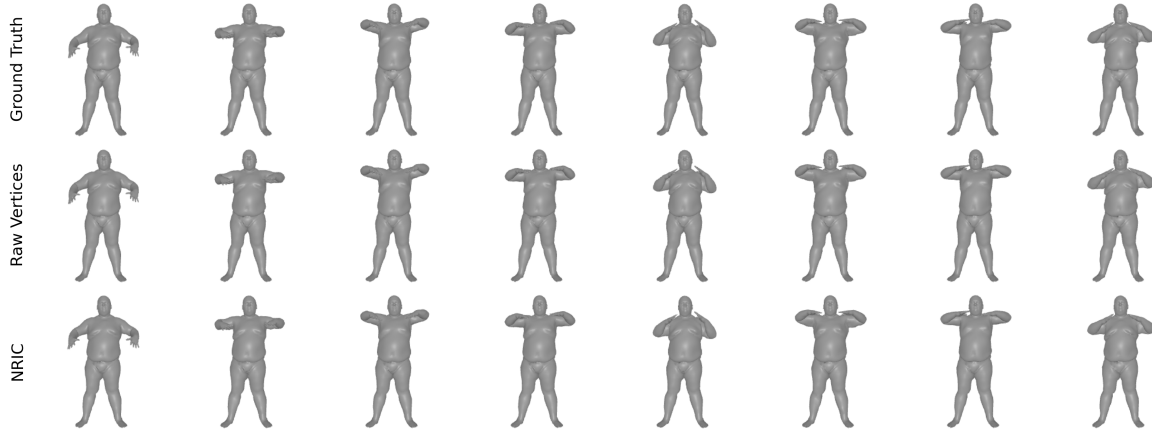


Figure 5.2: Comparison to ground-truth of the Chicken Wings sequence for both representations. The figure contains snapshots of the sequence every 8 frames.

5.2.3. Origin-Relative Feature Encoding

In this Experiment, rather than predicting the absolute feature values, the network is trained to predict deviations from the first frame of the sequence. For vertex coordinates, the target is the displacement $\Delta V^i = V^i - V^0$, recovered at evaluation by adding back the stored first frame. For NRIC, the targets are $\Delta L^i = L^i - L^0$ and $\Delta \theta^i = \theta^i - \theta^0$, recovered analogously before reconstruction. The motivation behind this experiment is to reduce the dynamic range of the regression targets, concentrating the network’s capacity on encoding deformation rather than absolute geometry, in hopes of giving NRIC an advantage. The results are available in Table 5.3, and a comparison between this and the previous encodings in Figure 5.3.

Origin-relative encoding improves both methods overall and narrows the gap between the models trained on raw vertex positions and those trained on NRIC. The improvement is not uniform across all

Sequence [#frames]	Raw vertex position		NRIC		
	Chamfer	NMSE	Chamfer	NMSE	NRIC Loss
Low Poly [48]	3.13×10^{-4}	7.20×10^{-9}	1.38×10^{-3}	3.10×10^{-7}	9.66×10^{-7}
High Poly [48]	1.28×10^{-4}	9.35×10^{-10}	2.01×10^{-3}	4.10×10^{-7}	3.66×10^{-7}
CW-02 [216]	1.78×10^{-3}	1.01×10^{-7}	6.72×10^{-3}	1.63×10^{-6}	3.99×10^{-6}
CW-04 [226]	2.26×10^{-3}	2.11×10^{-7}	7.82×10^{-3}	4.37×10^{-6}	1.07×10^{-5}
Punch-02 [303]	2.03×10^{-3}	1.32×10^{-7}	2.69×10^{-2}	3.42×10^{-5}	6.25×10^{-5}
Run-02 [331]	2.95×10^{-3}	2.36×10^{-7}	2.63×10^{-2}	4.17×10^{-5}	2.62×10^{-4}
Paper-C [107]	4.20×10^{-4}	1.30×10^{-6}	2.73×10^{-4}	1.04×10^{-6}	6.63×10^{-6}
Paper-S [107]	1.03×10^{-3}	1.07×10^{-6}	2.02×10^{-4}	1.22×10^{-7}	6.40×10^{-8}
Mean	1.37×10^{-3}	3.82×10^{-7}	8.96×10^{-3}	1.05×10^{-5}	4.34×10^{-5}

Table 5.3: Origin-Relative Feature Encoding results. Organized in the same manner as Table 5.2. Shows overall improvements for all sequences over all categories, most importantly in the 3D reconstruction of the paper sequences. Values are the mean over 3 runs, and the complete table containing the standard deviations can be found in Appendix A.2.

sequences, though, and one result goes against expectations.

For the synthetic sequences, raw vertices improve further into the negligible range, while NRIC remains consistent with the results of our previous experiment. As previously discussed, the cyclic nature of this motion may be to blame for this behaviour, the first frame not serving as a stable enough anchor for the NRIC features, meaning that the differences do not necessarily form a lower-variance signal than the absolute values themselves. The conventional approach does not seem to share this sensitivity and improves regardless of the cyclical nature.

Across DFAUST sequences, NRIC improves meaningfully in all cases. The sequence length pattern observed in the previous experiment persists but is less pronounced. All sequences manage to drop by an order of magnitude or more in NMSE, while almost halving the Chamfer error. Even with these improvements, the conventional approach stays consistently lower across all DFAUST sequences and metrics. The NRIC Loss values again point to reconstruction amplification, rather than poor feature prediction, as the primary source of the remaining gap.

The paper sequences once again favour NRIC, and the advantage becomes clearer under origin-related encoding. The paper strip in particular drops well below the perceptibility threshold for NRIC, while the corresponding raw vertex coordinates results stay above it. Origin-relative encoding removes the absolute geometric state of the paper, which has little bearing on the bending-dominated deformation signal. What remains is a target that NRIC’s intrinsic features can represent far more accurately, as reflected in the 10^{-8} magnitude of the NRIC Loss. While NRIC is the clear winner for this category, the conventional approach also improves, but proportionally less, remaining consistent with vertex coordinates benefiting less structurally from a deformation-relative anchor than the NRIC features do.

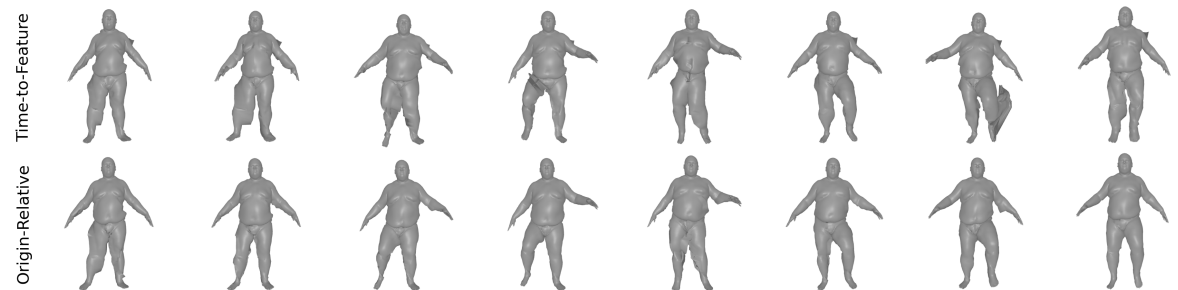


Figure 5.3: Snapshots of NRIC reconstructions of the Run-02 sequence comparing time-to-feature and origin-relative encodings. There is a clear improvement over time-to-feature in both quality of reconstruction and reduction of artefacts.

Median-Relative Variant

A variant of this experiment, median-relative encoding, replaces the first frame anchor with the frame whose features are closest to the per-feature median across the sequence, the one that minimises the average deviation from all other frames. Our motivation was that the first frame may not always represent a suitable rest configuration, and a more central anchor could reduce the variance of the different targets. In practice, as seen in Table 5.4, the results are a consistent downgrade across all sequences and both representations compared to our other experiments so far. The underlying issue is that a statistically centered frame places zero at the middle of the training sequence rather than at its boundary. The network must then learn a signal that grows away from zero, reaches its peak, and returns, or, in the case of non-cyclic sequences, one that oscillates on both sides of zero throughout. That is a harder function to fit than the monotonically increasing displacements that a rest-pose anchor tends to produce, where deviations from zero usually accumulate in one direction, and the network can exploit the temporal structure of the sequence.

Finding a geometrically meaningful anchor seems to matter more than finding a statistically central one, and for most of our sequences, the first frame serves that role better than the median frame; however, that is not to say that there is not a better anchor frame to be found within the sequence. With this in mind, the degradation is not symmetric. Models trained on raw vertex coordinates suffer more on several sequences, and this is most visible on the paper sequences, where their error rises sharply compared to the models trained on NRIC. Additionally, on most sequences other than paper, the NMSE converges to similar values for both representations, generally favouring NRIC as opposed to vertex coordinates. This variant of the Origin-Relative Feature Encoding is therefore an objectively worse approach in absolute terms, but one that incidentally narrows the gap between the two representations.

Sequence [#frames]	Raw vertex position		NRIC		
	Chamfer	NMSE	Chamfer	NMSE	NRIC Loss
Low Poly [48]	2.85×10^{-3}	5.32×10^{-5}	4.01×10^{-3}	2.64×10^{-5}	5.27×10^{-5}
High Poly [48]	2.83×10^{-3}	5.17×10^{-5}	3.62×10^{-3}	2.55×10^{-5}	3.07×10^{-5}
CW-02 [216]	3.70×10^{-3}	4.19×10^{-5}	1.31×10^{-2}	4.06×10^{-5}	2.31×10^{-4}
CW-04 [226]	3.89×10^{-3}	6.81×10^{-5}	1.60×10^{-2}	6.75×10^{-5}	2.11×10^{-4}
Punch-02 [303]	4.09×10^{-3}	1.21×10^{-5}	4.30×10^{-2}	1.20×10^{-4}	2.80×10^{-4}
Paper-C [107]	7.53×10^{-3}	2.67×10^{-2}	3.54×10^{-4}	4.74×10^{-5}	4.51×10^{-4}
Paper-S [107]	4.36×10^{-3}	2.62×10^{-3}	3.06×10^{-3}	2.14×10^{-3}	1.12×10^{-3}
Mean	4.04×10^{-3}	3.69×10^{-3}	1.45×10^{-2}	3.15×10^{-4}	3.32×10^{-4}

Table 5.4: Median-Relative Variant results. Does not contain the Run-02 dataset, as the general outcome of this experiment provided worse results than the alternatives. We see a consistent downgrade in accuracy across all metrics, categories, and sequences. Values are the mean over 3 runs, and the complete table containing the standard deviations can be found in Appendix A.3.

5.2.4. Interpolating Frames from Compressed Encoding

This experiment directly addresses the second research subquestion: whether a model trained to compress a mesh sequence can synthesize plausible intermediate frames. To answer this, we train the network on every other frame of the sequence (half the total frames), and evaluate exclusively on the frames that were skipped during training. If the network learns a smooth temporal mapping, querying it at an intermediate timestep should produce geometrically plausible reconstructions of the held-out frames.

Every result in Table 5.5 sits well above the 5×10^{-3} visible artefact threshold for Chamfer, confirming the expected finding that a network trained through overfitting does not generalize to intermediate timesteps. Because the model is optimized solely to minimize error on the training frames, it is free to fit a function that precisely passes through observed frames while behaving arbitrarily between them, which is what also occurs in practice. The premise of this experiment is therefore incompatible with the compression approach as applied here, and the results are presented primarily as the motivation for

Sequence [#frames]	Raw vertex position		NRIC		
	Chamfer	NMSE	Chamfer	NMSE	NRIC Loss
Low Poly [24]	4.15×10^{-2}	1.92×10^{-4}	5.00×10^{-2}	2.16×10^{-4}	9.08×10^{-4}
High Poly [24]	4.47×10^{-2}	2.07×10^{-4}	5.79×10^{-2}	2.85×10^{-4}	4.56×10^{-4}
CW-02 [108]	1.31×10^{-2}	7.51×10^{-6}	2.23×10^{-1}	1.57×10^{-3}	1.85×10^{-2}
CW-04 [113]	1.02×10^{-2}	6.43×10^{-6}	1.61×10^{-1}	1.34×10^{-3}	1.30×10^{-2}
Paper-C [53]	2.43×10^{-2}	6.60×10^{-3}	9.00×10^{-3}	9.58×10^{-4}	2.77×10^{-3}
Paper-S [53]	2.20×10^{-2}	5.01×10^{-4}	4.32×10^{-2}	5.80×10^{-3}	6.35×10^{-3}
Mean	2.60×10^{-2}	1.25×10^{-3}	9.07×10^{-2}	1.69×10^{-3}	7.00×10^{-3}

Table 5.5: Frame Interpolation results. Only contains a few examples for each category, primarily reducing the number of sequences tested from the DFAUST dataset. All results point to the general fact that overfitted networks do not handle data points outside their training well. Values are the mean over 3 runs, and the complete table containing the standard deviations can be found in Appendix A.4.

the second part of this research, the Variational Auto-Encoder.

5.2.5. Feature-to-Feature Mapping

All previous experiments provide the network with a scalar timestep and train it to predict the corresponding mesh state. This section explores an alternative formulation in which the network receives the current frame as input and is trained to predict the next frame in the sequence. Rather than learning a mapping from the time to geometry, the model learns a mapping from one geometric state to the next, operating purely in feature space without any temporal context.

This change has two architectural consequences for the architecture. First, the network no longer takes a scalar as input but a full feature vector, namely a flattened vertex matrix for the conventional approach and a concatenated vector of edge lengths and dihedral angles for NRIC. The input layer therefore scales with the size of the mesh, which is substantially larger than the single-neuron input used in previous examples and results in a considerably larger network overall. Second, the Fourier time encoding is no longer applicable, as there is no temporal signal to embed. The network receives organic features directly, losing the structured frequency decomposition that helped the MLP resolve fine-grained temporal variation in earlier experiments. The remainder of the architecture remains unchanged and is kept consistent with previous experiments. A diagram of the new architecture is shown in Figure 5.4.

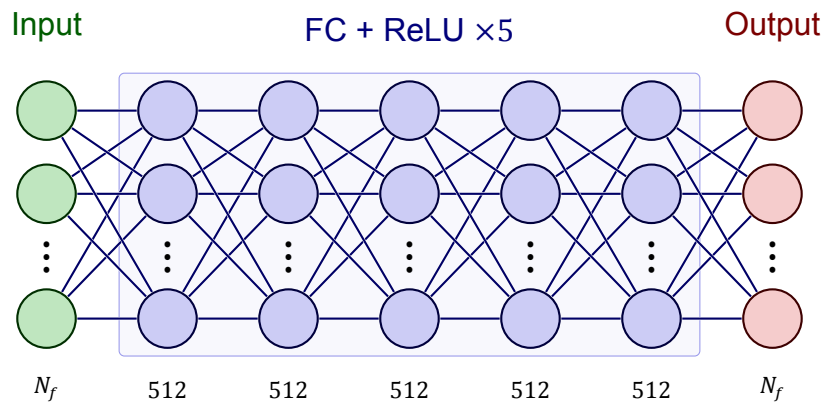


Figure 5.4: *FrameNet* variant architecture. The network receives features for frame i of the sequence and predicts the features of frame $i + 1$. N_f represents the number of output features: $N_f = |V| \times 3$ for raw vertex positions, and $N_f = |E| \times 2$ for NRIC features.

We use two evaluation methods for this experiment:

- The **Normal** evaluation, where the ground-truth frame is used as input at each step to predict the following frame, isolating the single-step prediction error.
- The **Cumulative** evaluation, where the network’s own prediction at each step is used as input for the next, simulating actual sequential reconstruction, but also allowing errors to propagate across the sequence.

Sequence [#frames]	Normal					Cumulative				
	Conv		NRIC			Conv		NRIC		
	Chamfer	NMSE	Chamfer	NMSE	NRIC L.	Chamfer	NMSE	Chamfer	NMSE	NRIC L.
Low Poly [48]	3.06×10^{-2}	1.64×10^{-4}	4.10×10^{-2}	1.31×10^{-4}	4.67×10^{-4}	1.22×10^{-1}	3.07×10^{-3}	6.12×10^{-1}	2.04×10^{-2}	3.19×10^{-1}
CW-02 [216]	9.95×10^{-3}	3.05×10^{-6}	2.49×10^{-2}	2.69×10^{-5}	4.94×10^{-5}	1.45×10^{-1}	1.61×10^{-3}	8.93×10^{-1}	1.98×10^{-2}	1.29×10^{-1}
Paper-S [107]	2.50×10^{-3}	5.03×10^{-6}	3.18×10^{-3}	9.59×10^{-5}	1.22×10^{-4}	9.42×10^{-2}	1.59×10^{-2}	2.16×10^{-1}	5.83×10^{-2}	4.05×10^{-2}
Mean	1.44×10^{-2}	5.74×10^{-5}	2.30×10^{-2}	8.48×10^{-5}	2.13×10^{-4}	1.20×10^{-1}	6.84×10^{-3}	5.74×10^{-1}	3.28×10^{-2}	1.63×10^{-1}

Table 5.6: Feature-to-Feature mapping results for normal and cumulative evaluation. Normal evaluation uses ground-truth frames as input for predicting the next frame and measures the difference between the prediction and the actual next frame. Cumulative evaluation, which is better suited for the pure purpose of compression as it only requires the first frame, uses the previously predicted frame to propagate the entire reconstruction of the sequence. Both evaluations reveal that this approach is not suitable for either representations or the sequences we experiment on. Values are the mean over 3 runs, and the complete table containing the standard deviations can be found in Appendix A.5.

Results for both evaluations are displayed in Table 5.6. Under normal evaluation, the frame-to-frame model produces acceptable results for the paper sequence, with Chamfer distances approaching those of the time-based experiment. On the DFAUST and synthetic sequences, however, the results are substantially worse, well above the threshold for visible artefacts in both metrics. Under cumulative evaluation, the results degrade rapidly across all sequences and representations, reaching the highest errors in all experiments so far.

The poor results come from two issues that compound each other. The first is the loss of the Fourier encoding, which in the earlier experiments was essential for helping the MLP resolve high-frequency temporal structure. Without it, the network has to learn a complex non-linear mapping from a high-dimensional input, and it does so without the structured inductive bias that the frequency decomposition used to provide. The second and more fundamental issue is that many of the sequences we use in the experiments are cyclic or near-cyclic, having consecutive frames that look nearly identical, which may require the network to predict completely different next frames depending on the direction of the motion. This creates an inherently ambiguous, multi-valued mapping that a deterministic MLP cannot resolve. The cumulative evaluation amplifies this, as any error in a prediction is passed forward as the input for each step, causing not only errors to accumulate throughout the sequence, but also encountering the same difficulty discovered in Section 5.2.4 of producing data outside the boundaries of overfitting by inputting never-before-seen frames into the network and expecting a plausible prediction. Moreover, by using the ground-truth frames as input for predicting future frames in the normal evaluation, we are foregoing the purpose of this application, compressing a sequence of meshes, and through the need of every original mesh in the reconstruction process, we gain no improvement on the compression of the sequence. Thus, the feature-to-feature formulation is not a viable alternative to the time-indexed approach under our conditions of study.

5.2.6. Discussion

Taken together, the compression experiments point to a consistent conclusion. For most of the sequences and experiment types we consider, conventional vertex coordinates offer more benefits for mesh sequence compression through overfitting than NRIC does. The gap is clearest on the DFAUST sequences, where NRIC trails by several orders of magnitude in NMSE and produces Chamfer distances above the perceptible artefact threshold. On the synthetic sequences, both representations reach effectively negligible error, which leaves the comparison there inconclusive, though raw vertex coordinates still come out ahead thanks to their smaller feature size. NRIC shows a clear and consistent advantage only on the paper crumpling sequences.

The reason for this pattern is that NRIC’s structural benefits, the locality of deformation and the invariance to rigid-body transformations are most valuable when the deformation is concentrated in bending rather than stretching, and when the motion does not repeat. Paper crumpling meets both

conditions. The edge lengths remain mostly constant throughout the sequence, the deformation generally resides in the dihedral angles, and the motion moves directly from one geometric state to another without repeating. Here, NRIC’s encoding is naturally aligned with the structure of the data, and the network has less total variation to fit across the two output channels. DFAUST human motion is the opposite. It involves cyclic articulations with large global displacement, body-surface dynamics that change many edges simultaneously, and longer sequences that place greater demand on the model’s capacity. These properties work against NRIC’s local sparsity, and the larger feature space, with roughly twice as many features as vertex coordinates for the same mesh, only adds to the difficulty without bringing a structural benefit.

RQ1: Where does NRIC’s error originate, and how does this change with mesh resolution?

NRIC’s error has two parts, and only the first is shared with the conventional approach. The first part is the prediction error of the model itself, which, for the vertex coordinates approach, is the only error we evaluate. For NRIC, the error begins at this prediction step and then grows once the features are reconstructed back into 3D geometry. Whether this reconstruction is direct or carried out by a least squares solver, the mesh we evaluate in 3D space always carries a higher relative error than the features predicted in NRIC space.

Two structural problems drive this. The first is the density of the representation. As previously mentioned, NRIC describes the same mesh with roughly twice as many features as vertex coordinates, so the network needs more capacity to take in the larger input, comes under more pressure to compress it, and has to learn to produce the larger output as well. The overhead is present no matter how well the deformation suits the representation. The second problem is the reconstruction step. The NRIC loss values throughout the experiments show that the network’s feature prediction is usually accurate, so the gap in both 3D Chamfer distance and NMSE traces back to reconstruction, where small errors in the predicted features are amplified into much larger errors in 3D space. This amplification grows with mesh resolution and deformation complexity, and it compounds across the stages of the pipeline, which also require feature normalization and Kabsch post-processing. Conventional coordinates carry none of this overhead and produce 3D geometry directly.

The effect of resolution is also not the same for every mesh. At low resolution, the two representations stay closely matched, and aside from the overhead NRIC carries, neither holds a meaningful edge. As resolution increases, the outcome appears to depend on the type of mesh the models are learning. For the DFAUST sequences, the added resolution works against NRIC and turns it into an overall disadvantage, since the reconstruction error grows faster than any benefit the representation provides. For the paper crumpling sequence, the opposite happens, and NRIC performs better even at the higher resolution, because its encoding remains well aligned with the bending-dominated deformation, regardless of how many edges are involved.

RQ2: Can an overfitted model synthesise plausible intermediate frames, and is it limited by the approach or the representation?

The second research subquestion asks whether a model trained to compress a mesh sequence can synthesise plausible intermediate frames. The frame interpolation experiment in Section 5.2.4 gives a direct answer, which is no. Because the network is trained purely by overfitting, it is only constrained on the frames it sees during training and is free to behave arbitrarily between them, so querying it at an intermediate timestep produces arbitrary output rather than a plausible interpolation. This holds for both representations, which shows that the limitation comes from the overfitting approach itself and not from the choice of coordinate system. The answer is therefore clearly negative, but it is a useful one. It is exactly this limitation that pushed our research towards a different approach and motivated the variational autoencoder studied in the following chapter.

5.3. Latent Space and Mesh Synthesis

Where the compression experiments asked how efficiently a neural network can memorize a single mesh sequence, the latent space experiments ask a fundamentally different question: whether a model trained across a collection of meshes can learn a continuous low-dimensional space in which every mesh is a point, nearby points correspond to geometrically similar meshes, and new points can be decoded into plausible geometries that were not present in training. This shift from pre-sequence memorization to cross-collection generalization is the central challenge of the second task, and the

properties we evaluate reflect it. We are no longer primarily interested in reconstruction fidelity as a compression measure, but in the structure and navigability of the learned space itself.

5.3.1. Experimental Setup

Each experiment trains a single VAE on a full mesh collection, following the architecture and training procedure described in Section 4.2. Unlike the compression experiments, which were averaged over 3 runs, our VAE implementation is more expensive to run, and, as such, we report metrics of single experiments instead. Although only one experiment cannot directly be used to draw conclusions consistently, our interpretations of the results are borne out of the many runs used to calibrate and optimize the hyperparameters for both representations, using the reported metrics as examples of the performance of the representations.

Datasets. The three dataset configurations used across the latent space experiments differ in scale, diversity, and the type of variation they present for the model.

- **MANO dataset.** The first is the MANO right hand dataset [18], consisting of 895 meshes of the human right hand, each representing a distinct pose drawn from the MANO model space and containing approximately 700 vertices each. The hand meshes carry no pose category label, making this dataset more suitable for evaluating the intrinsic structure of the learned latent space without reference to any ground truth grouping. As described in Section 4.2.2, all meshes are aligned to a globally consistent wrist orientation before training. The relatively low vertex count and constrained anatomical variation make this the easiest to analyze of the three datasets, and it serves as the primary setting in which the hyperparameter requirements of each representation are established before scaling to larger body datasets.
- **Single DFAUST subject.** The second dataset consists of all motion sequences of a single DFAUST subject, subject 50002, whose some individual sequences were also used in the compression experiments. Rather than treating these as separate sequences to be compressed independently, all meshes across 14 motions are pooled into a single flat collection of approximately 4,700 instances, each labeled by the motion sequence it belongs to. This labeling is not used during training but is used at evaluation time to assess whether the latent space organizes itself in a way that reflects the motion categories present in the data. The fixed body shape across all meshes means that any variation in the latent space must reflect differences in pose and motion, with no body identity to account for.
- **Multiple DFAUST subjects.** The third and largest dataset extends this to five subjects from the DFAUST dataset: 50002, 50004, 50007, 50009, and 50020, each performing up to 14 motion sequences for a total of approximately 19,000 meshes. Each mesh carries two labels: its motion category and the identity of the subject it belongs to. The introduction of multiple subjects adds body shape variation that was absent in the single-subject setting, creating a more demanding test of what each representation encodes in its latent space and allowing us to ask whether the space separates subjects from motions.

Hyperparameter configuration. Although the VAE architecture is kept identical across both representations, as described in Section 4.2.2, the two representations differ in structure and in their preprocessing requirements. As such, it is not possible to use the exact same hyperparameters for both.

The main difference between the two configurations comes from how the decoder output activation interacts with the variational objective. Since the NRIC features are normalized to the range $[-0.9, 0.9]$ before training, the decoder uses a Tanh activation to match that range. This bounds the reconstructions and keeps gradients flowing through the encoder throughout training. The conventional approach has no such bounded activation, because raw vertex coordinates are not confined to a fixed range. At the standard initialization of $\mu = 0$ and $\log \sigma^2 = 0$, the KL gradient with respect to $\log \sigma^2$ evaluates to $-0.5(1 - \exp(0)) = 0$, so without a bounded decoder activation to keep the encoder gradients alive, the conventional approach depends on the free bits mechanism to provide a non-zero gradient signal. For this reason, it uses a slightly higher free bits threshold than NRIC on the smaller datasets, and NRIC is usually trained with a smaller maximum KL weight. Besides these two adjustments, both of

which follow mostly from the difference in activation, the hyperparameters are kept consistent across the two representations.

5.3.2. Reconstruction Quality

The standard reconstruction metrics, mean Chamfer distance and NMSE, are carried over directly from the compression experiments and computed on a held-out test split of each dataset. As in Section 5.2.1, Chamfer distance is the preferred primary metric for its robustness to isolated vertex outliers and its independence from strict vertex correspondence. Table 5.7 reports results across all three datasets.

Dataset	Vertex Coords		NRIC	
	Chamfer	NMSE	Chamfer	NMSE
Right Hand	6.47×10^{-3}	9.81×10^{-5}	5.49×10^{-3}	1.01×10^{-4}
DFAUST 50002	8.85×10^{-3}	1.80×10^{-6}	3.90×10^{-2}	6.26×10^{-5}
DFAUST Multi-subject	1.10×10^{-2}	3.42×10^{-6}	6.74×10^{-2}	1.97×10^{-4}
Mean	8.77×10^{-3}	3.45×10^{-5}	3.73×10^{-2}	1.20×10^{-4}

Table 5.7: Reconstruction results for a random sample of 25% of the training sets in Chamfer distance and NMSE for both representations. Samples are seeded and present the same meshes between representations.

On the right hand dataset, NRIC achieves a marginally lower Chamfer distance while the conventional approach achieves a marginally lower NMSE, leaving the two effectively level at this scale, which still contrasts with the consistent conventional advantage seen in the compression experiments. The least-squares reconstruction introduces little additional error here, and the normalized NRIC features provide a well-conditioned regression target that the decoder can fit accurately. On the body datasets, the conventional approach pulls clearly ahead on both metrics, and the gap grows with dataset size. The growing body shape diversity increases the range of deformation types the NRIC decoder must represent, and small feature prediction errors are amplified as mesh complexity increases. While these types of results are expected because of our experiments with sequence compression, as previously discussed, the intention for our VAE is not replication of inputs, but comparison in how each latent space organizes information as well as how it produces new information. As such, we hold the metrics that follow this section to a higher importance than just reconstruction accuracy.

5.3.3. Dimension Analysis

The variational objective pushes the encoder to spread meaningful information across the latent dimensions, but in practice the model resists this pressure when the reconstruction loss dominates or when the free bits floor suppresses KL gradients. To understand how a trained model actually uses its latent space, we look at how the KL divergence values are distributed across the dimensions. The Gini coefficient tells us how uneven this distribution is across dimensions, in which a value of zero means that every dimension contributes equally, while a value near one means the contribution is concentrated in just a few dimensions. The standard deviation of the per-dimension KL values, kl_{std} , indicates whether the free bits floor has suppressed the KL gradient across all dimensions, as a near-zero value means that the latent organization is driven mostly by reconstruction pressure. We also report the number of dimensions whose per-dimension KL exceeds 0.5, which gives a direct count of the active dimensions above the free bits floor. Although the KL values for each dimension show how active they are and how the encoder learned to express the data in the latent space, they say nothing about how these dimensions function together and whether dimensions are independent or correlated with each other. To verify this, we report the participation ratio of the encoded means of our training sets, $\frac{(\sum \lambda)^2}{\sum \lambda^2}$, where λ are the eigenvalues of their covariance. This metric tells us how many dimensions the data actually covers, unaffected by correlations between them. The results are available in Table 5.8.

In all three datasets, the conventional approach holds onto an active latent space that sits well above the free bits floor. Its kl_{std} comes out highest in the table for every setting, and the number of

Dataset	Vertex Coords				NRIC			
	Gini	KL Std	Dims > 0.5	Eff. Dim	Gini	KL Std	Dims > 0.5	Eff. Dim
Right Hand (24 dims)	0.281	1.121	6/24	3.7	0.385	0.343	24/24	14.1
DFAUST 50002 (48 dims)	0.369	1.496	17/48	11.4	0.501	0.250	48/48	22.1
DFAUST Multi-subject (64 dims)	0.260	1.884	29/64	14.1	0.453	0.812	62/64	25.8

Table 5.8: Table portraying the dimension analysis for both representations. The table holds the Gini values, standard deviation, the number of dimensions with KL values over 0.5, and the number of effective dimensions used (participation ratio) for all three datasets.

dimensions above 0.5 climbs as the dataset grows in size. This tells us that the space is shaped by the variational objective rather than by reconstruction pressure on its own. Its fairly low Gini shows that the KL is distributed quite evenly over the active dimensions, making it a generally consistent space when traversing it.

NRIC organizes its space in a completely different way. Almost every dimension climbs above 0.5 on all three datasets, so the representation makes use of nearly its entire available latent space. In contrast, its kl_{std} is more moderate, and its Gini is higher than the conventional approach in every case. This points to information that is spread less evenly, where some dimensions specialize and carry more, while others stay closer to the prior. The participation ratio supports this as well, being roughly two to four times higher on NRIC than for the vertex coordinate approach. It also suggests that, even though most dimensions are active for both representations, the conventional representation encodes data in a far smaller effective subspace, and that NRIC distributes them in a more independent manner. This confirms that NRIC’s higher dimensionality coverage is not redundant, as dimensions generally hold different data, independently from others.

Both representations therefore keep an actively organized latent space, and the real difference between them is how each distributes its data. The conventional approach encodes its dimensions broadly and evenly, while NRIC makes sharper distinctions between them. This is likely to be a direct result of the representation, where conventional coordinates represent a single feature in 3D, and NRIC is constructed from 2 separate features, allowing this distinction in dimensions to happen.

5.3.4. Latent-to-Geometry Correlation

To assess how faithfully the latent space reflects 3D geometric similarity, we compute the Spearman correlation between the pairwise latent distances and the pairwise mean per-vertex Euclidean distances between the corresponding reconstructed meshes, evaluated over a fixed random subset of the training set. We refer to this as latent versus 3D correlation. Because two input representations do not encode the same amount of 3D geometric information, we also compute the same correlation directly between the pairwise raw input feature distances and the pairwise 3D distances. This provides a feature ceiling against which each model’s correlation can be expressed as a percentage. The ratio enables a direct comparison of how efficiently each representation uses the geometric information available to it, independent of the differing absolute ceiling values, to organize data based on closeness in 3D space. Results are displayed in Table 5.9 and the PCA’s of the feature and latent spaces for both representations in Figure 5.5.

In absolute terms, the conventional approach achieves a higher latent versus 3D correlation on the body datasets, while on the right hand dataset, NRIC is marginally ahead. Vertex coordinates are directly predictive of per-vertex Euclidean distance, so a space organized around them tends to correlate well with 3D geometry, which makes the ordering on the body datasets expected. The more informative comparison is against each representation’s own ceiling, since NRIC’s ceiling is structurally lower as a direct consequence of its rotation invariance: two meshes differing only in global orientation are identical in NRIC space yet carry a non-zero per-vertex Euclidean distance in 3D, which lowers the theoretical maximum that any NRIC model can reach.

Measured against these ceilings, efficiency varies by dataset rather than following a single trend. On the right hand dataset, NRIC uses a substantially larger share of its available signal than vertex coordinates, while on the single-subject DFAUST set, the ordering reverses in favour of vertex coordinates. The multi-subject case is the most notable, as there the NRIC latent correlation passes its own feature ceiling, meaning that the learned encoding aligns latent geometry with 3D shape geometry

Dataset	Vertex Coords			NRIC		
	Corr.	Ceiling	%	Corr.	Ceiling	%
Right Hand	0.395	0.975	40.5	0.437	0.528	82.6
DFAUST 50002	0.694	0.962	72.1	0.317	0.661	47.9
DFAUST Multi-subject	0.794	0.955	83.1	0.446	0.347	128.4

Table 5.9: The correlation between meshes encoded in latent space and the same meshes in 3D space. The ceilings represent the similarity between how each of the input features is organized in comparison to a 3D space. The ceiling for conventional vertex coordinates is almost 1, as it organizes meshes under the same criteria as itself, the 3D space, while the ceilings for NRIC show how the NRIC representations themselves directly correlate with their 3D counterparts. Each correlation is also displayed as a percentage of its ceiling.

more tightly than the raw feature distances correlate with the per-vertex Euclidean distances in 3D, and a learned non-linear encoder is not bound by it. The encoder can re-weight and recombine the intrinsic features into a latent metric that tracks 3D similarity better than the raw feature distance can. The diversity of the multi-subject collection likely supplies enough signal to the encoder to find such a combination, producing a latent space whose distances follow 3D shape geometry better than the features it is built from.

5.3.5. Latent Space Traversal

The most direct way to understand a latent space is by traversing it. We vary each latent dimension across a fixed range while holding all others at zero, and we then analyze the decoded mesh sequences for geometric change. Mean displacement per step measures the average Chamfer change between successive decoded meshes along a traversal. The decoder coefficient of variation measures how much the decoder output norms vary across a full traversal grid. A high value means that the dimensions produce changes of comparable magnitude. For the NRIC models, mode purity measures how strongly each dimension’s decoder weights concentrate on either the edge length channel or the dihedral angle channel, which then points to specialization towards a bending or a stretching deformation. Along with these metrics, we examine the highest-scoring dimensions by hand to judge whether their geometric effect is interpretable and anatomically plausible, since the aggregate metrics on their own cannot say whether a dimension controls something meaningful. The results are displayed in Table 5.10.

Dataset	Vertex Coords		NRIC		
	Displ.	Decoder CV	Displ.	Decoder CV	Mode Purity
Right Hand	0.009	1.652	0.025	0.266	0.120
DFAUST 50002	0.039	1.488	0.043	0.206	0.017
DFAUST Multi-subject	0.041	1.467	0.086	0.448	0.167

Table 5.10: Table contains the mean displacement in the geometry of a mesh per dimension and the coefficient of variation in the decoder for both vertex coordinates and NRIC, with the addition of mode purity, which measures the mean ratio between bending and stretching in NRIC dimensions.

On the right hand dataset, NRIC dimensions produce mean displacements nearly three times larger per unit latent step than their conventional counterparts. Traversing the conventional latent space produces barely perceptible changes to the hand geometry even at large step sizes, while equivalent or smaller steps along NRIC dimensions produce directed, anatomically recognizable changes in pose. This is illustrated directly by Figure 5.6, which shows latent space explorations for both representations on the right hand dataset.

Read together with the mean displacements, the decoder coefficient of variation reveals a clear difference in how each space distributes its expressive power. NRIC moves the geometry further per

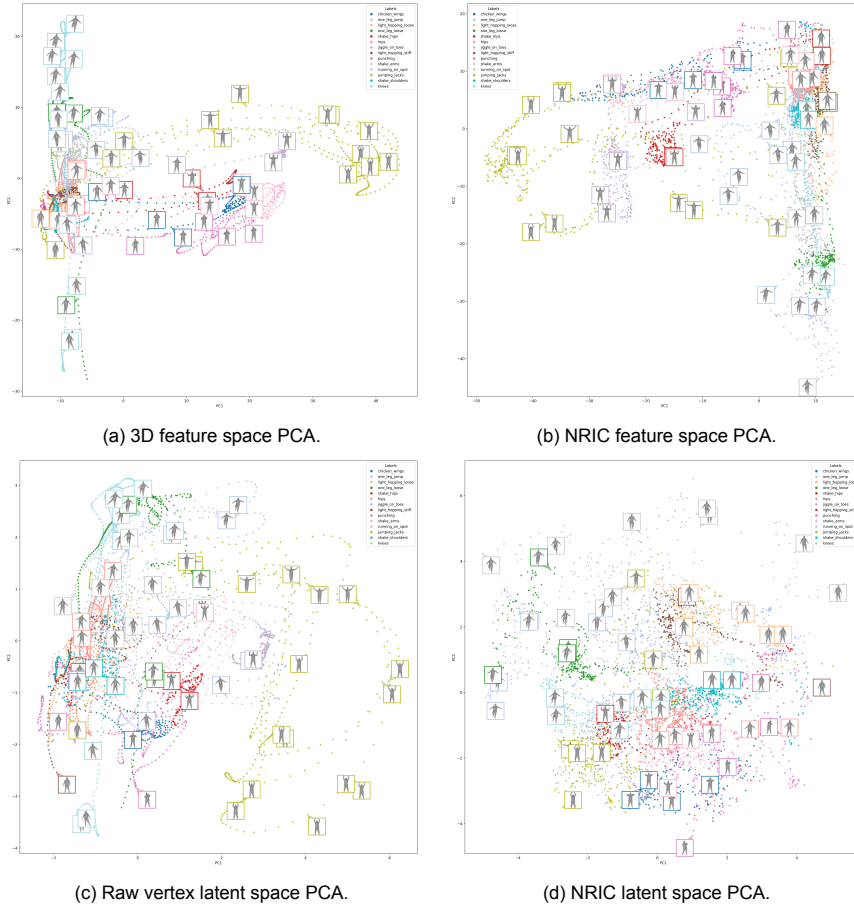


Figure 5.5: Feature versus latent space comparison between 3D and NRIC representations of the DFAUST single-subject dataset. All plots are labeled with color based on the motion sequence each mesh belongs to.

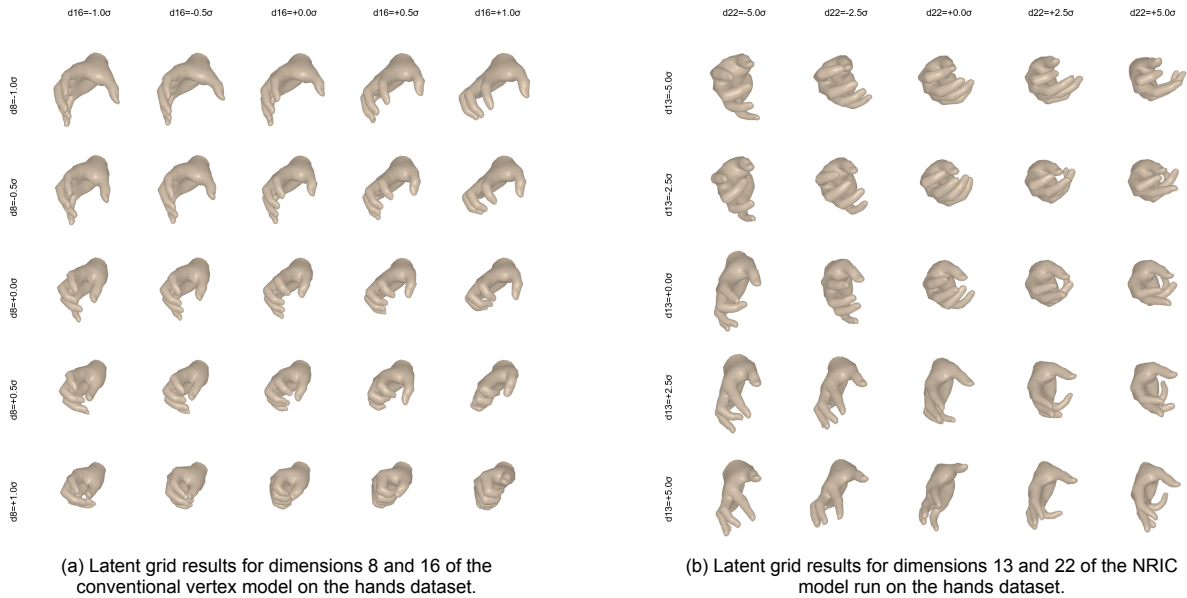


Figure 5.6: Latent space manual exploration on dimensions with highest impact and change for both NRIC and conventional approach for the right hands dataset.

latent step and keeps the magnitude similar no matter which dimension is traversed, so its low variability reflects a space in which nearly every dimension is live and contributes a meaningful change. The conventional vertex coordinate approach displays the complete opposite behaviour. It pairs a smaller average displacement with higher variability, meaning that most of its dimensions reshape the mesh only slightly, while a small number of them carry almost all of the visible change. Its expressive power therefore sits in a few strongly active dimensions, with the rest left largely inert and unused. NRIC, by comparison, spreads directed and comparably large change across the whole of its latent space. This contrast holds across all three datasets.

Finally, the mode purity results do not show the model learning to separate bending from stretching in any consistent way. Some imbalance does appear in certain dimensions across the dataset, where they lean towards one channel or the other, hinting that a change in the training objective could lead the encoder to separate the two more deliberately. For now, though, the specialization is not steady or pronounced enough across the settings to support a more exact claim, so we present it as a possibility rather than a result.

5.3.6. Interpolation and Generation

Following the evaluation approach of Tan et al. [22], we assess shape interpolation by linearly blending the latent codes of pairs of held-out meshes and decoding the intermediate points, and shape generation by sampling from the standard normal prior and decoding the resulting latent vectors. For interpolation, we use two metrics. The latent path ratio compares the length of the interpolated path to the direct distance between the endpoints in latent space, in which a value close to one indicates a near-linear path. The continuity difference measures how smooth the decoded mesh sequence is, in terms of how much change there is between consecutive frames, with lower values meaning more gradual transitions. This captures how consistent the path of interpolation stays in latent space, but does not measure the validity of the path. For mesh generation we again rely on two metrics. The nearest-neighbour distance is the average Chamfer distance from each decoded sample to its closest training mesh, and the diversity ratio is the fraction of distinct training meshes that appear as nearest neighbours across the generated samples. Results for both are shown in Table 5.11.

Dataset	Vertex Coords				NRIC			
	Path Ratio	Continuity	NN Dist.	Diversity	Path Ratio	Continuity	NN Dist.	Diversity
Right Hand	1.351	0.22	0.012	0.950	1.536	7.51	0.018	0.850
DFAUST 50002	1.055	1.78	0.067	0.950	1.323	16.13	0.151	0.950
DFAUST Multi-subject	1.024	1.84	0.082	0.900	1.419	18.36	0.250	0.950

Table 5.11: Interpolation and generation results for 8 distinct interpolation target pairs and 30 generated meshes from latent space for each dataset.

The conventional approach gives mathematically smoother interpolations overall, in the sense that the average change between interpolated frames is small and even. The reason for this is its uniformly organized posterior, which holds the interpolated latent codes near the training distribution along the entire path. NRIC interpolations have a higher continuity difference, and on the body datasets, the least-squares reconstruction turns small inconsistencies in feature space into visible 3D transitions that do not always take a direct path between the two endpoints. This kind of smoothness, however, does not tell the full story. Since the conventional approach interpolates directly between vertex positions in 3D space, the intermediate meshes are not guaranteed to be geometrically plausible, something that can be seen in Figure 5.7. Fingers can shrink or compress as vertices move linearly towards their next positions, which produces local volume artefacts that match no realistic hand configuration. On the other hand, NRIC avoids this completely by construction, as interpolation of edge lengths and dihedral angles preserves the intrinsic geometry of the surface at every step, keeping the volume consistent and the geometry plausible. Figure 5.8 shows example interpolation sequences for both representations, confirming that, while more mathematically smooth, the motion presented by raw vertices is direct and unnatural, and that the motion of NRIC is more natural thanks purely to its representation of lengths and angles.

Generation results show the conventional approach generating samples that sit closer to the training distribution across all datasets, reflected in its consistently lower nearest-neighbour distance. Diversity

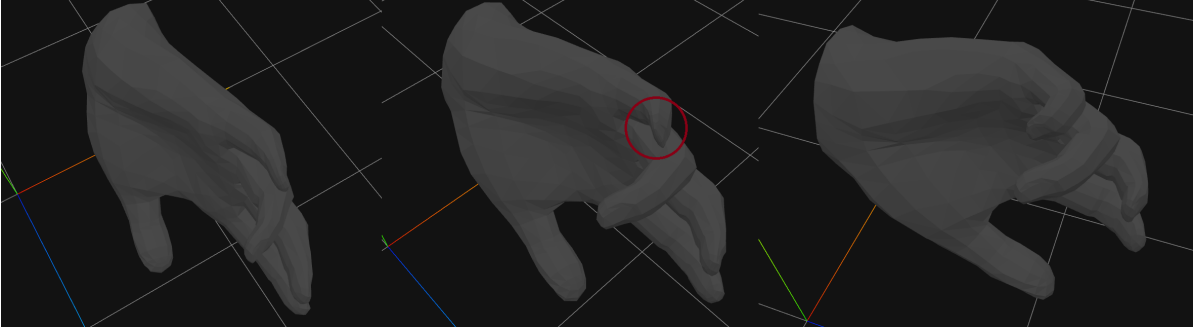


Figure 5.7: Volume shrinkage of finger during hand model interpolation in latent space.

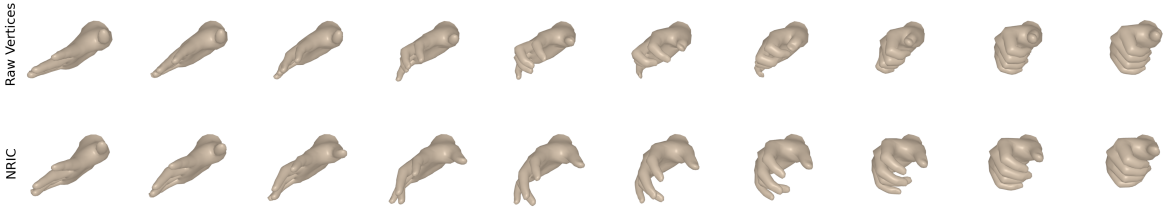
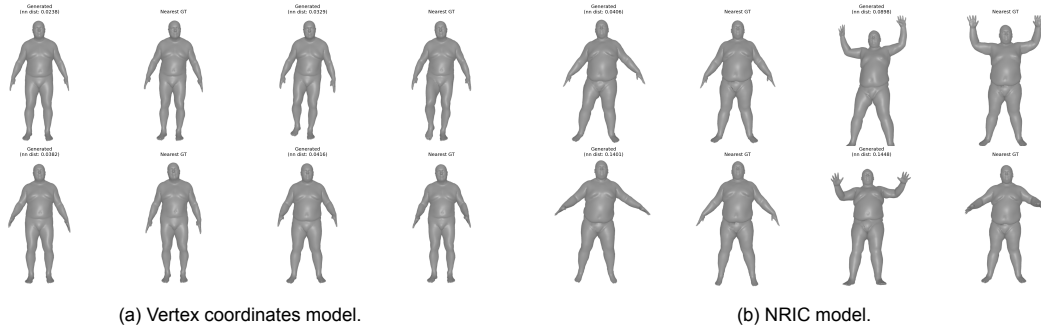


Figure 5.8: Interpolation in latent space of hand dataset model trained on both representations and between the same endpoints. We can see that NRIC provides a more geometrically sound interpolation, as the movement is more plausible, when compared to the vertex coordinates interpolation.

is comparable between the two representations rather than favouring one, and, considering the chosen sample size of 30 random meshes, the difference in results can be considered negligible. Overall, both spaces are able to synthesize new and plausible meshes, even when randomly sampled, showing that with some direction, any mesh that can be inferred from the training sets could be produced. Some generation results for the DFAUST single-subject dataset can be seen in Figure 5.9.



(a) Vertex coordinates model.

(b) NRIC model.

Figure 5.9: Generation of meshes and their nearest-neighbour from models of both representations and of the DFAUST 50002 subject.

5.3.7. Class Separation

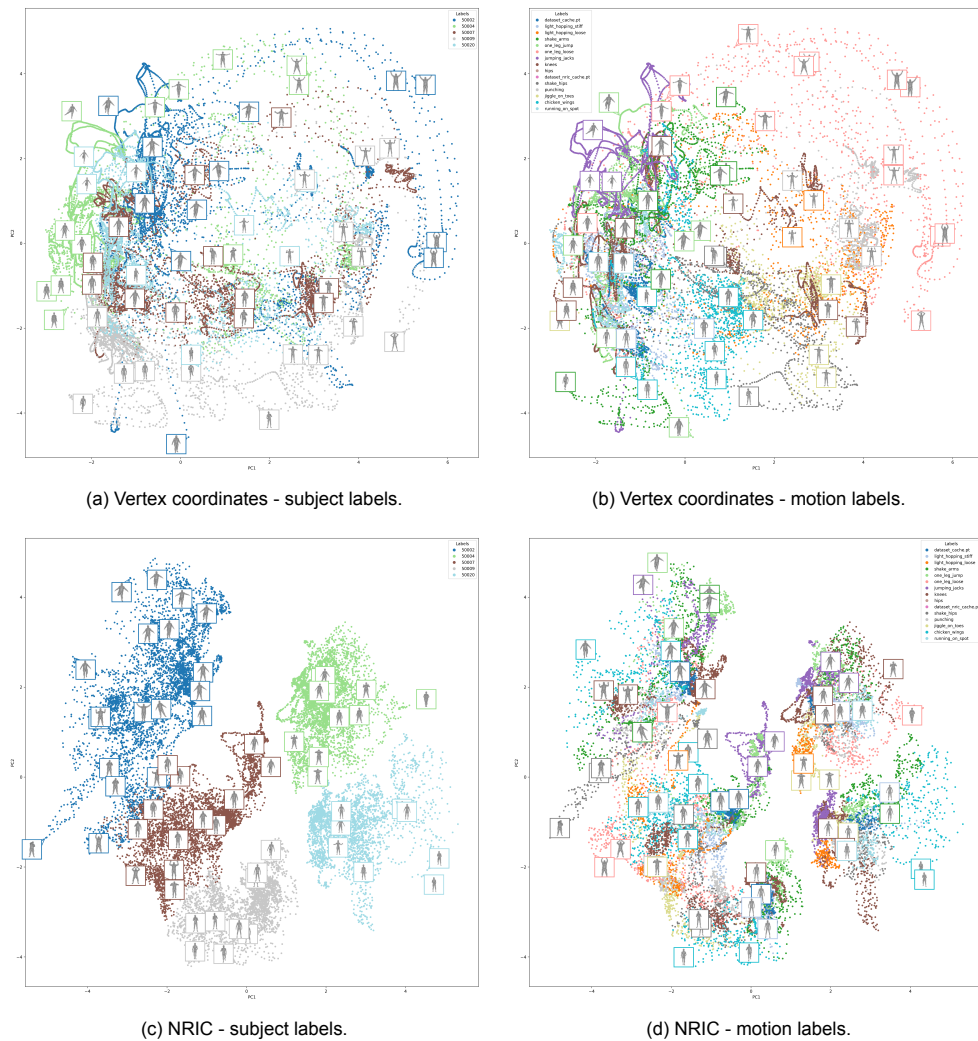
For the labeled DFAUST datasets, we evaluate latent space organization using three measures applied separately to motion type and subject identity labels. The silhouette score measures the ratio of class cohesion to between-class separation, ranging from minus one to one. The inter-to-intra class distance ratio indicates whether different classes occupy distinct regions even when their boundaries are not clean. Finally, KNN accuracy measures what fraction of test points have their nearest latent neighbour in the same class, serving as a ceiling reference for local separability independent of global cluster structure. Results are shown in Table 5.12.

Across all three datasets, the aggregate separation metrics are modest for both representations, and on their own, they do not clearly favour one over the other. On the motion labels, the conventional

Labels	Conv			NRIC		
	Silhouette	Inter/Intra	kNN	Silhouette	Inter/Intra	kNN
DFAUST 50002 — Poses	0.067	1.264	1.000	0.045	1.221	1.000
DFAUST Multi-subj. — Poses	-0.020	1.209	1.000	-0.021	1.123	1.000
DFAUST Multi-subj. — Subjects	0.074	1.134	1.000	0.109	1.176	1.000

Table 5.12: Class separation metrics for the labeled DFAUST datasets. Silhouette measures the ratio of class cohesion to between-class separation, and the inter-to-intra ratio measures whether different classes fall into distinct areas of the space. Additionally, the kNN score is a consistent one across the board, making it negligible for our comparison, but a good sign of local clustering.

approach holds a slight silhouette advantage, while on the subject labels, NRIC gives a more separated space. KNN accuracy reaches its ceiling of 1 on all configurations, confirming that both latent spaces have a consistent local clustering structure, regardless of the two other metrics. These measurements are each computed over a single label at a time and across the whole collection at once, which leaves them blind to any nested structure that organizes one factor inside another. The principal component projections (PCA) in Figure 5.10 reveal exactly that kind of structure, and they tell a much clearer story than the metrics on their own.



On the multi-subject dataset, the NRIC latent space separates the subjects directly. Each subject sits in its own distinct region of the space, resolved cleanly even in the top two principal dimensions, with little overlap between identities. The conventional space, however, is not arranged this way, at least not in any discernible way from its PCA, and so it leaves the subjects intermingled rather than settled into separate regions. This is the direction the subject label metrics point to, since NRIC reaches the higher silhouette and inter-to-intra ratio, but the projection shows the effect far better than any scalar.

The organization in NRIC space goes one step further, as it appears to have arranged itself in a hierarchical manner. Within each separated subject region, the individual motions stay visually distinguishable, so that the space is able to first resolve which subject a mesh belongs to and then, locally, how that subject is moving. This nested alignment is also the reason why NRIC's flat motion silhouette looks unremarkable. When the same motion is spread across several subjects rather than gathered into one compact cluster, a metric that looks at motion labels alone and at all subjects at once, one cannot properly evaluate the within-subject separation that the projection makes obvious. In contrast, the conventional space shows no comparable nesting, preferring to organize both motion and subject in path-like structures, possibly originating from the ordered form of the meshes within an animation. Looking at the PCA of its space, some structure exists, but it becomes difficult to draw hard lines to separate either label type, making it harder to identify the type of motion or the type of subject performing said motion for most positions in this space.

5.3.8. Discussion

Unlike the sequence compression chapter, the VAE experiments are not as interested in how accurately each representation can be restructured. The question here is what kind of space originates from each representation and is judged on four properties: how well it is organized, how easily it can be navigated, how its points are distributed, and how usable it is for generation. Because no single metric captures all of this, we rely on a broad set of evaluation methods, each targeting a different aspect of the space. Reconstruction still matters in a supporting role, since a space is only worth analyzing if the model behind it is trained well enough to be trusted. Even so, reconstruction is not the property we are comparing, and where a gap does appear between the two representations, it does not need a new explanation. The previous chapter has already shown why vertex coordinates decode more accurately than NRIC, and the same reasoning carries over to this setting, even with the shift to least squares reconstruction.

Once reconstruction is set aside, the remaining metrics divide the two representations cleanly, with each holding advantages the other does not. The reason for the split lies in what each representation encodes in the first place. Conventional coordinates store the absolute position of every vertex in 3D space, so a latent space built from them tends to mirror 3D geometry and to keep the encoded points close to where each mesh sits. NRIC stores intrinsic deformation, the edge lengths and dihedral angles that describe the surface itself, so a latent space built from it is organized more by how it changes, not by where it is positioned. The two spaces are therefore measuring similarity along different ideas of what makes two meshes alike, and most of the contrast that follows is a consequence of that difference.

RQ3: Does NRIC organize the latent space more meaningfully? The answer is yes, and three separate metrics support it. The *KL* dimension analysis shows that NRIC makes use of almost all of its available dimensions, and it does so with a higher Gini coefficient on every dataset. We read the uneven spread as a healthy sign, since it means that some dimensions can specialize and carry information more meaningfully than the rest. The conventional space is active as well, but it never makes full use of the latent dimensions, leaving more than half of them at the free bits floor. The ones that are active share their *KL* evenly, so no single dimension functions as the one responsible for large geometric changes. Besides using almost all of its available dimensions, NRIC also spreads them independently, its participation ratio reflecting that most of these dimensions contribute differently to the encoding of the latent space. On the other hand, even with fewer active dimensions, the conventional approach's encodings are concentrated into a low effective dimensionality, so its latent space's expressive capacity rests on a few independent axes.

The latent-to-geometry correlation supports the same conclusion. NRIC's ceiling is lower than the conventional one by design, since the two meshes that differ only in orientation are identical in NRIC space, but distinct in 3D, so part of the 3D variation is invisible to it from the start. Even with this lower ceiling, NRIC generally fills more of it than the conventional approach fills its own, and on the datasets

with enough variation, the NRIC space reaches past its ceiling, organizing itself closer to 3D geometry than the raw NRIC features manage on their own.

Finally, the class separation results reinforce this. The scalar metrics do not point to a clear advantage for either representation, but the PCA visualizations do. They show that NRIC space forms globally distinct clusters for each subject in the training set, and within each of those clusters, it groups the individual motions again, so the space is organized on two levels at once. The conventional approach cannot draw the same hard boundaries between subjects, which leaves open the question of whether fidelity to 3D space is worth more than a more intrinsic form of organization.

RQ4: How do the two spaces differ for synthesis and navigation? The interpolation and generation results do not point towards a clear winner. Conventional vertex coordinates reconstruct in a direct manner, moving vertices evenly from start to target as previously discussed, making the motion sometimes feel unnatural. Moreover, these transitions are not always geometrically plausible, and since vertices move along straight lines towards their targets with nothing to hold the surface together in between, the fingers of the hand models can lose volume during interpolation, and the arms and hands of the DFAUST models can shrink in the same way. NRIC does not interpolate in this manner. By construction, it blends edge lengths and dihedral angles rather than vertex positions, and since those quantities describe the surface itself, the volumes stay intact, the artefacts are avoided altogether, and, while not as consistent in jumps between interpolations, the motion stays natural and plausible. While this explains how both representations handle interpolation, the models themselves also follow the same principles. Interpolating within the space and then decoding the mesh, similar behaviour can be observed, making it a consequence of representation.

For generation, both representations produce plausible meshes when sampled from the prior, and their diversity is comparable. NRIC's advantage here lies in how the space is organized rather than in the quality of the samples, since its layout makes it easier to target regions that correspond to specific subjects and poses, as the PCA of its latent shape shows. That same organization also shapes how each space is traversed. When the goal is controlled synthesis, NRIC is the stronger option because its structure gives clear regions to steer towards. When the goal is instead to move through a space that behaves like the 3D features used in the conventional approach, NRIC offers less benefit.

Which space is more useful for synthesis comes down to the priorities of the task. For direct and consistent blending that stays close to the training distribution, the conventional approach is the better choice. For geometrically valid and natural output that can be steered by subject or motion, NRIC is the better starting point.

6

Conclusion

This thesis investigated whether Nonlinear Rotation-Invariant Coordinates can be more beneficial for machine learning applications on 3D meshes than conventional raw vertex coordinates. We explored this through two machine learning tasks: mesh sequence (animation) compression through overfitting, as well as latent space learning with a variational autoencoder, run across datasets ranging from simple synthetic sequences to large collections of human body motion. This chapter brings the findings together and answers each of the research questions from the introduction, closing with the central question that motivated this research.

RQ1: Where does NRIC’s error originate, and how does this change with mesh resolution? For compression through overfitting, conventional vertex coordinates are the more reliable of the two input representations on most of the sequences we tested. NRIC’s performance is several orders of magnitude below the conventional representation in NMSE on the DFAUST sequences, and it pulls clearly ahead only on the paper crumpling data, where the bending-dominated, non-repeating deformation matches what the representation captures best. During training, the NRIC loss remains low, so the network generally predicts the intrinsic features accurately. The rest of the gap in 3D quality appears at the reconstruction step, where small errors in predicted edge lengths and dihedral angles grow into much larger errors in 3D space. There are additional aspects, specific only to the NRIC pipeline, which add to this error: the density of the same mesh with about twice as many features, and the pipeline overhead of normalization, GOAST reconstruction, and Kabsch alignment that vertex coordinates do not require.

Other than that, resolution does not act uniformly either, because at low resolution, the two representations stay closely matched, but as the resolution of the mesh rises, the outcome starts depending on the type of mesh. On the DFAUST sequences, the added resolution amplifies the reconstruction error and becomes an overall disadvantage for NRIC, while on the paper crumpling sequence, NRIC holds its advantage even at higher resolutions, since its encoding stays aligned with the deformation, being less affected by the increase in mesh size.

RQ2: Can an overfitted model synthesize plausible intermediate frames? The straight answer is no. A network trained purely by overfitting is only constrained by the frames it sees during training, so querying it at an intermediate timestep returns arbitrary output rather than a plausible interpolation. Both representations fail in the same way, which shows that the limitation belongs to the overfitting approach itself and not to the choice of coordinate system. The answer is clearly negative, but it is also a useful one, since this limitation is exactly what motivated us to move to the variational autoencoder in the second part of this thesis.

RQ3: Does NRIC organize the latent space more meaningfully? It does. The KL dimension analysis shows that NRIC engages almost all of its available dimensions and does so with a higher Gini coefficient on every dataset, which means that some dimensions can specialize and carry more information than the rest, as well as doing so across more independent dimensions, since its participation

ratio is double the conventional one at equal latent size. The conventional space is active too, but it leaves more than half of its dimensions at the free bits floor and shares its KL evenly among the others, so none of the dimensions stands out in particular. The latent-to-geometry correlation tells a similar story. NRIC’s ceiling is structurally lower because NRIC is, by definition, rotation invariant, and so meshes that differ only by rigid body transformations differ only in a 3D space.

Regardless, NRIC fills more of its own ceiling than the conventional approach does. On the multi-subject set, its latent metric passes that ceiling entirely, reflecting 3D shape similarity more closely than raw NRIC features do. The clearest evidence is in the class separation that NRIC achieves. Although the scalar metrics favour neither representation, the PCA projections show the NRIC space forming distinct clusters for each subject, with the individual motions grouped again inside each cluster, meaning that the space is organized on two levels at once, without having access to labels during training. The conventional space cannot draw these boundaries as well and leaves the subjects mixed together in unclear ways.

RQ4: How do the two spaces differ for synthesis and navigation? Interpolation and generation showcase a trade-off between the representations. Conventional coordinates reconstruct better, and consecutive steps within the interpolation path change the mesh evenly. That consistency, however, is not a guarantee of plausible geometry, because moving vertex positions along a straight line can shrink the volume of fingers or limbs in the interpolated result, resulting in an unnatural movement. Similarly, the latent space originating from vertex positions also displays the same behaviour. NRIC avoids this by construction, because blending edge lengths and dihedral angles holds the intrinsic geometry of the surface intact at every step, thus keeping the movement through interpolation natural.

For navigation, NRIC’s subject-structured layout makes it easier to steer towards a particular subject or motion, which suits controlled synthesis, while the conventional space is the better fit when the goal is to move through a space that behaves more like the input space of raw 3D features. Which space is more useful comes down to the priority of the task.

RQ5: Do NRIC’s advantages generalize across tasks? No, all experiments point to these advantages being task-specific, and the two halves of this thesis prove it in different ways. In compression, NRIC carries a fixed overhead from its larger feature space and its reconstruction step, and this overhead does the most damage when the task asks the model to reconstruct a cyclical sequence, since both the reconstruction error and the cyclic nature of the sequence amplify the error. The representation only pays off on the crumpling paper sequence, where the deformation of the meshes suits NRIC.

In the latent space setting, reconstruction stops being the deciding factor, and the intrinsic organization of NRIC turns into a benefit, making a more structured and traversable space with geometric consistency during synthesis that vertex coordinates are not guaranteed to provide. The pattern across both tasks is consistent. NRIC proves most beneficial when the model has to generalize structure across large and varied collections, and least beneficial when it has to memorize repeating sequences of frames where bending and stretching happens all across the mesh surface. NRIC’s value highly depends on the task at hand, as well as the material it is meant to work with.

The central question. Does NRIC prove more beneficial in machine learning applications than conventional raw vertex coordinates? In a general sense, the answer is no, but it has the potential to be under the right conditions. For sequence compression, vertex coordinates, or the representations made from them, such as RIMD, are the better choice for most data, and NRIC wins only in the specific case of bending-dominated, non-repeating deformations. For latent space learning, NRIC produces spaces with more visible structure, better transferability, and some geometric guarantees during synthesis that vertex coordinates cannot match. However, this comes at a lower reconstruction fidelity on complex datasets and with more pipeline complexity throughout, even with the advantage of directly processing collections with large rotational and positional differences in the meshes.

The two representations are not really competing for the same thing. Vertex coordinates are numerically convenient in their nature and geometrically transparent. NRIC captures what the geometry is doing rather than where it is, and when a task can use that organizational structure, it produces results that vertex coordinates cannot replicate. Whether this is worth the additional cost depends entirely on the task at hand and the goal of the application.

NRIC should not be treated as a drop-in replacement for vertex coordinates in general mesh learning pipelines, but as a representation that earns its overhead when the task matches its strengths. Intrinsic deformations, rotation invariance where it genuinely helps a dataset or mesh type, and generalization across diverse collections rather than memorization of a fixed sequence are areas where NRIC can be stronger than alternatives. It is in those settings that NRIC's advantages are real and distinct.

Reflection

This thesis produced results that were, in some places, surprising, and it is worth stepping back to assess not just what was found but how the work was conducted, where it falls short, and what a more complete investigation could look like.

7.1. Limitations

The first point concerns the main methodological decision in both tasks, which was to keep the architecture and training procedure as identical as possible across the two representations, so that any difference in the results could be attributed to the coordinate system rather than the model. The principle is sound and gives us interpretable comparisons, but it also builds in an asymmetry that should be acknowledged. NRIC describes the same mesh with roughly twice as many features as vertex coordinates, so the output layer of every network in the compression experiments is about twice as large for NRIC. The architecture was not adjusted with this in mind, which left the model capacity completely unchanged for a harder regression problem. This choice kept the comparison clean, but it likely contributed to NRIC's weaker results on the DFAUST sequences. A completely architecture-neutral comparison might scale the capacity to the size of the output rather than fixing it, but even so, how would one evaluate whether the scaling applied to NRIC is truly bringing it to an equal footing with the other representation?

More fundamentally, the reconstruction bottleneck is a cost that training cannot fix. Throughout the compression experiments, the NRIC loss values showed that the network's predictions were reasonably accurate and that part of the gap between NRIC and vertex coordinates in 3D Chamfer distance came directly from the reconstruction step rather than from poor feature learning. This means that even with a perfect prediction, NRIC would still carry some reconstruction overhead. Any future work using NRIC in a compression context needs to reckon with this directly, either by greatly reducing the error in NRIC space or by designing the evaluation metrics that do not penalize it.

The compression task, as implemented in this thesis, is a controlled proof of concept rather than a working compression system. Training one network per sequence for 20,000 iterations and comparing the accuracy of the predictions is a fair way to measure the learnability of each representation, but it lacks the analysis of resulting storage sizes or comparisons with other mesh compression formats. A stronger framing would treat the compression component as a measurement of its own. This would also raise the question of whether overfitting is the right approach at all. A model that generalizes across the frames of a sequence, learning the structure of the motion instead of memorizing single timesteps, would make a more practical compressor and a fairer test of whether NRIC's smoother deformation paths really help a network learn. Neural ODEs [3] or recurrent networks trained to predict the next state of a sequence are worth exploring on this front, and they might give NRIC more room to show its structural advantages over a longer sequence.

Moreover, the VAE architecture used in this thesis is a flat MLP encoder and decoder that treats the input as an unstructured vector. This was the right choice for a representation-neutral comparison, but it leaves a large amount of geometric structure unexploited in both representations. Graph neural networks or spectral convolution architectures that operate directly on mesh connectivity could, in

principle, give the encoder access to the local neighbourhood relationships already present in NRIC's edge structure and might produce a qualitatively different kind of latent space. Testing NRIC with a topology-aware encoder is a natural step that this thesis did not take.

The choice for the VAE learning objective is also worth reflecting on. The KL divergence penalty, even with the free bits threshold, creates a tension between reconstruction quality and latent space organization that was originally difficult to resolve for vertex coordinates. A Wasserstein autoencoder that replaces the KL term with a maximum mean discrepancy, enforces the aggregate prior without turning each individual posterior towards a Gaussian. This could allow not only the conventional approach to develop a more organized structure in its latent space, but also help in realizing a more meaningful and traversable space for both representations.

Another design choice for the VAE was to use no label supervision during training. The DFAUST datasets carry motion and subject identity labels that were used only at evaluation time. A conditional or semi-supervised setup that uses these labels to organize the latent space during training, similar to the constrained synthesis of Tan et al. with RIMD features [22], could reveal whether NRIC's more distributed latent structure is easier or harder to steer towards meaningful organization than the conventional approach's spatially separated paths.

7.2. Wider Context and Implications

Placing the limitations aside, this thesis asks a question that the field has yet to answer, which is where the burden in the choice of structure should rest. Some works put it in the design of the architectures, building rotation and translation invariance into the network through spectral or spatial convolutions, as explored by Bronstein et al. and implemented in methods such as MeshCNN and DiffusionNet. NRIC puts the same burden on the data, so that even a plain MLP operates on quantities that are invariant by construction. Our results find that neither side has it entirely right. A vertex-based model with enough capacity can still produce decent results, even if it is not as complex as the state-of-the-art approaches, and even surpasses NRIC on raw fidelity. However, the latent spaces it learns are less structured than those NRIC produces under the same architecture, which is evidence that a representation can carry work an architecture would otherwise do, albeit in specific scenarios. The compression task also generally supports this idea, since training a small network to reproduce a single sequence is itself a form of implicit neural representation, the same concept now used to compress images and volumes.

These connections suggest where the representation could be useful in practice. The clearest opening comes from the paper crumpling result. Cloth and other thin-shell structures deform in much the same way, which is exactly the setting where NRIC became more beneficial than raw vertex coordinates. Compression of cloth animation and latent space synthesis of draping or folding are, therefore, the applications most likely to reward the representation, and a focused follow-up study would start with a strong prior in its favour. A second direction is deformable shape completion. Litany et al. complete partial meshes with graph convolutional autoencoders that work on vertex data [13], and our results raise the question of whether the same task is easier in an intrinsic representation, where a missing region is described by local edges and dihedral angles rather than by absolute positions that have to be inferred from a global frame. A third and sharper direction is comparison with RIMD. RIMD is the rotation-invariant representation closest in spirit to NRIC, and Tan et al. report large gains from it over raw coordinates [22]. We compare NRIC against vertex coordinates, but never against other representations such as RIMD, which leaves the fairly informative question of which intrinsic representation organizes a latent space more meaningfully. We decided, however, to keep the scope of this thesis constrained and only focus on the most basic comparisons, thus serving as possible building blocks for future research.

The work done in this thesis also carries some risks that are worth stating directly. The first is tied to what the latent space turned out to encode. The class separation results showed that NRIC space splits subjects into distinct regions, which means that the representation makes individual body identity easy to recover. The property that helps the models organize a dataset is the same one that would aid in re-identifying a person from their body shape, so that any use of intrinsic mesh representations on human capture data raises a privacy concern. The second is a lesser risk that considers the over-generalization of the interpretations made throughout this thesis. Our conclusions rest on human bodies, hands, and a sheet of paper, and presenting NRIC as broadly better than it is, based solely on the strength of the results in the paper, could invite misuse in settings where its assumptions do not fit.

These risks and applications have different effects on different groups. The most direct stakeholders are other researchers in the field of geometric deep learning or working with intrinsic representations like NRIC, who gain a controlled comparison to build on rather than a new model to adopt. Next are the possible users who would feel the trade-offs of NRIC in use, among them are animation and visual effect artists who compress and edit meshes, and the engineers who stream animated geometry for games, TV, and more. Another group is the developers who would have to turn NRIC into a usable tool, where the work would focus on making the reconstruction step faster to run and more widely available, as, in its current state, it is computationally heavy and locked behind the GOAST library rather than exposed in a form that most pipelines can use.

Read together, these threads set the thesis inside a wider effort in geometric deep learning to work out whether structure is best imposed through the architecture or through the representation. The dominant approach has been architectural, building invariances and local to global hierarchies into the model itself. NRIC takes the alternative, that the right representation makes the architecture simpler because the structure the model needs to learn is already in the data. Our results here suggest that both positions hold something real. Conventional coordinates with a complex enough architecture can produce competitive results, as seen in most state-of-the-art approaches, but the latent spaces they learn seem less structured and less geometrically interpretable than those NRIC produces under the same architecture. Whether using NRIC is worth the reconstruction overhead that comes with it depends solely on the application, and this thesis does not answer that question in the general sense. What it offers, however, is a controlled, empirical comparison of its trade-offs that future works can build upon.

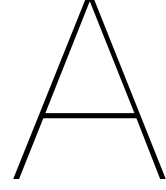
7.3. Use of AI

AI was used in several ways over the course of this thesis. The most significant was as a tool for brainstorming ideas, both for implementing the models and for evaluating and visualizing their results. Even so, no AI-written code was included in the thesis without proper assessment and verification, since it is not reliable enough to be entrusted directly with implementation. Beyond this, AI supported the writing of the paper itself by easily transferring data from the JSON results produced by our experiments into Overleaf-ready markdown tables and by helping to understand and build figures with the TikZ library. Finally, since most spell-checking tools today are based on AI, such programs were used to check the spelling flow and correctness of the text throughout.

Bibliography

- [1] Federica Bogo et al. “Dynamic FAUST: Registering Human Bodies in Motion”. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. July 2017.
- [2] Michael M. Bronstein et al. “Geometric Deep Learning: Going beyond Euclidean data”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42. ISSN: 1558-0792. DOI: 10.1109/msp.2017.2693418. URL: <http://dx.doi.org/10.1109/MSP.2017.2693418>.
- [3] Ricky T. Q. Chen et al. *Neural Ordinary Differential Equations*. 2019. arXiv: 1806.07366 [cs.LG]. URL: <https://arxiv.org/abs/1806.07366>.
- [4] Xin Chen et al. “Executing your Commands via Motion Diffusion in Latent Space”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 18000–18010.
- [5] Marvin Eisenberger et al. *NeuroMorph: Unsupervised Shape Interpolation and Correspondence in One Go*. 2021. arXiv: 2106.09431 [cs.CV]. URL: <https://arxiv.org/abs/2106.09431>.
- [6] Indefin Gaming. URL: https://www.turbosquid.com/3d-models/animated-paper-crumpling-3d-1794996?dd_referrer=.
- [7] Yulan Guo et al. *Deep Learning for 3D Point Clouds: A Survey*. 2020. arXiv: 1912.12033 [cs.CV]. URL: <https://arxiv.org/abs/1912.12033>.
- [8] Rana Hanocka et al. “MeshCNN: a network with an edge”. In: *ACM Transactions on Graphics* 38.4 (2019), pp. 1–12. ISSN: 1557-7368. DOI: 10.1145/3306346.3322959. URL: <http://dx.doi.org/10.1145/3306346.3322959>.
- [9] Florine Hartwig et al. *Geodesic Calculus on Implicitly Defined Latent Manifolds*. 2026. arXiv: 2510.09468 [cs.LG]. URL: <https://arxiv.org/abs/2510.09468>.
- [10] Behrend Heeren, Josua Sassen, et al. *The Geometric Optimization And Simulation Toolbox*. 2020. URL: <https://gitlab.com/numod/goast>.
- [11] Alex Kendall, Yarin Gal, and Roberto Cipolla. *Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics*. 2018. arXiv: 1705.07115 [cs.CV]. URL: <https://arxiv.org/abs/1705.07115>.
- [12] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: <https://arxiv.org/abs/1412.6980>.
- [13] Or Litany et al. *Deformable Shape Completion with Graph Convolutional Autoencoders*. 2018. arXiv: 1712.00268 [cs.CV]. URL: <https://arxiv.org/abs/1712.00268>.
- [14] Arman Maesumi et al. “PoissonNet: A Local-Global Approach for Learning on Surfaces”. In: (2025).
- [15] mujib81. URL: <https://free3d.com/3d-model/lowpoly-rigged-femalemale-for-game-77955.html>.
- [16] NVIDIA. *CUDA Technology* <http://www.nvidia.com/CUDA>. 2007.
- [17] Nasim Rahaman et al. *On the Spectral Bias of Neural Networks*. 2019. arXiv: 1806.08734 [stat.ML]. URL: <https://arxiv.org/abs/1806.08734>.
- [18] Javier Romero, Dimitrios Tzionas, and Michael J. Black. “Embodied Hands: Modeling and Capturing Hands and Bodies Together”. In: *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*. 245:1–245:17 36.6 (Nov. 2017).

- [19] Josua Sassen, Klaus Hildebrandt, and Martin Rumpf. “Nonlinear Deformation Synthesis via Sparse Principal Geodesic Analysis”. In: *Computer Graphics Forum* 39.5 (2020), pp. 119–132. DOI: <https://doi.org/10.1111/cgf.14073>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14073>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14073>.
- [20] Josua Sassen et al. “Geometric optimization using nonlinear rotation-invariant coordinates”. In: *Computer Aided Geometric Design* 77 (Feb. 2020), p. 101829. ISSN: 0167-8396. DOI: 10.1016/j.cagd.2020.101829. URL: <http://dx.doi.org/10.1016/j.cagd.2020.101829>.
- [21] Nicholas Sharp et al. *DiffusionNet: Discretization Agnostic Learning on Surfaces*. 2022. arXiv: 2012.00888 [cs.CV]. URL: <https://arxiv.org/abs/2012.00888>.
- [22] Qingyang Tan et al. *Variational Autoencoders for Deforming 3D Mesh Models*. 2018. arXiv: 1709.04307 [cs.GR]. URL: <https://arxiv.org/abs/1709.04307>.
- [23] W. Jakob, J. Rhineland, D. Moldovan and others. *pybind11 – Seamless operability between C++11 and Python*. 2017.
- [24] T. Winkler et al. “Multi-Scale Geometry Interpolation”. In: *Computer Graphics Forum* 29.2 (2010), pp. 309–318. DOI: <https://doi.org/10.1111/j.1467-8659.2009.01600.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01600.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01600.x>.
- [25] Yi Zhou et al. *Fully Convolutional Mesh Autoencoder using Efficient Spatially Varying Kernels*. 2020. arXiv: 2006.04325 [cs.CV]. URL: <https://arxiv.org/abs/2006.04325>.



Appendix A

Sequence [#frames]	Raw vertex position		NRIC		
	Chamfer	NMSE	Chamfer	NMSE	NRIC Loss
Low Poly [48]	$5.80 \times 10^{-4} \pm 1.71 \times 10^{-4}$	$1.85 \times 10^{-8} \pm 1.02 \times 10^{-8}$	$1.84 \times 10^{-3} \pm 1.86 \times 10^{-3}$	$3.36 \times 10^{-7} \pm 4.48 \times 10^{-7}$	$1.44 \times 10^{-6} \pm 1.93 \times 10^{-6}$
High Poly [48]	$4.07 \times 10^{-4} \pm 4.34 \times 10^{-5}$	$8.96 \times 10^{-9} \pm 1.82 \times 10^{-9}$	$9.07 \times 10^{-4} \pm 1.50 \times 10^{-4}$	$5.64 \times 10^{-8} \pm 1.63 \times 10^{-8}$	$1.07 \times 10^{-7} \pm 3.09 \times 10^{-8}$
CW-02 [216]	$2.73 \times 10^{-3} \pm 5.63 \times 10^{-4}$	$2.28 \times 10^{-7} \pm 8.50 \times 10^{-8}$	$1.33 \times 10^{-2} \pm 4.81 \times 10^{-3}$	$1.23 \times 10^{-5} \pm 5.47 \times 10^{-6}$	$1.86 \times 10^{-6} \pm 1.85 \times 10^{-6}$
CW-04 [216]	$2.46 \times 10^{-3} \pm 4.84 \times 10^{-6}$	$2.43 \times 10^{-7} \pm 2.32 \times 10^{-9}$	$1.32 \times 10^{-2} \pm 5.71 \times 10^{-3}$	$1.00 \times 10^{-5} \pm 5.68 \times 10^{-6}$	$1.29 \times 10^{-5} \pm 1.29 \times 10^{-5}$
Punch-02 [303]	$2.64 \times 10^{-3} \pm 2.83 \times 10^{-5}$	$2.16 \times 10^{-7} \pm 3.09 \times 10^{-9}$	$4.69 \times 10^{-2} \pm 1.31 \times 10^{-2}$	$1.60 \times 10^{-4} \pm 1.41 \times 10^{-4}$	$5.02 \times 10^{-5} \pm 1.85 \times 10^{-5}$
Run-02 [331]	$4.07 \times 10^{-3} \pm 9.85 \times 10^{-4}$	$3.86 \times 10^{-7} \pm 1.30 \times 10^{-7}$	$4.83 \times 10^{-2} \pm 4.90 \times 10^{-3}$	$1.33 \times 10^{-4} \pm 4.95 \times 10^{-5}$	$1.46 \times 10^{-4} \pm 2.44 \times 10^{-5}$
Paper-C [107]	$7.69 \times 10^{-4} \pm 5.35 \times 10^{-4}$	$5.12 \times 10^{-6} \pm 5.60 \times 10^{-6}$	$5.01 \times 10^{-4} \pm 6.68 \times 10^{-4}$	$7.64 \times 10^{-6} \pm 1.08 \times 10^{-5}$	$6.06 \times 10^{-6} \pm 8.52 \times 10^{-6}$
Paper-S [107]	$2.35 \times 10^{-3} \pm 1.87 \times 10^{-3}$	$3.39 \times 10^{-6} \pm 3.42 \times 10^{-6}$	$8.80 \times 10^{-4} \pm 3.70 \times 10^{-4}$	$7.26 \times 10^{-5} \pm 1.01 \times 10^{-4}$	$9.10 \times 10^{-6} \pm 2.60 \times 10^{-6}$
Mean	$2.09 \times 10^{-3} \pm 1.23 \times 10^{-3}$	$1.09 \times 10^{-6} \pm 1.85 \times 10^{-6}$	$1.48 \times 10^{-2} \pm 1.93 \times 10^{-2}$	$4.42 \times 10^{-5} \pm 6.25 \times 10^{-5}$	$2.56 \times 10^{-5} \pm 4.79 \times 10^{-5}$

Table A.1: Complete time-to-feature table, presenting both the results and the variance of the data over the 3 runs.

Sequence [#frames]	Raw vertex position		NRIC		
	Chamfer	NMSE	Chamfer	NMSE	NRIC Loss
Low Poly [48]	$3.13 \times 10^{-4} \pm 1.22 \times 10^{-4}$	$7.20 \times 10^{-9} \pm 4.28 \times 10^{-9}$	$1.38 \times 10^{-3} \pm 1.62 \times 10^{-3}$	$3.10 \times 10^{-7} \pm 4.32 \times 10^{-7}$	$9.66 \times 10^{-7} \pm 1.33 \times 10^{-6}$
High Poly [48]	$1.28 \times 10^{-4} \pm 1.49 \times 10^{-5}$	$9.35 \times 10^{-10} \pm 1.85 \times 10^{-10}$	$2.01 \times 10^{-3} \pm 6.17 \times 10^{-4}$	$4.10 \times 10^{-7} \pm 2.19 \times 10^{-7}$	$3.66 \times 10^{-7} \pm 2.56 \times 10^{-7}$
CW-02 [216]	$1.78 \times 10^{-3} \pm 3.35 \times 10^{-6}$	$1.01 \times 10^{-7} \pm 5.44 \times 10^{-10}$	$6.72 \times 10^{-3} \pm 8.65 \times 10^{-4}$	$1.63 \times 10^{-6} \pm 2.27 \times 10^{-7}$	$3.99 \times 10^{-6} \pm 1.65 \times 10^{-6}$
CW-04 [226]	$2.26 \times 10^{-3} \pm 1.13 \times 10^{-4}$	$2.11 \times 10^{-7} \pm 2.43 \times 10^{-8}$	$7.82 \times 10^{-3} \pm 4.60 \times 10^{-3}$	$4.37 \times 10^{-6} \pm 1.85 \times 10^{-6}$	$1.07 \times 10^{-5} \pm 1.03 \times 10^{-5}$
Punch-02 [303]	$2.03 \times 10^{-3} \pm 7.90 \times 10^{-6}$	$1.32 \times 10^{-7} \pm 8.99 \times 10^{-11}$	$2.69 \times 10^{-2} \pm 1.43 \times 10^{-3}$	$3.42 \times 10^{-5} \pm 8.11 \times 10^{-6}$	$6.25 \times 10^{-5} \pm 9.81 \times 10^{-7}$
Run-02 [331]	$2.95 \times 10^{-3} \pm 1.06 \times 10^{-3}$	$2.36 \times 10^{-7} \pm 1.43 \times 10^{-7}$	$2.63 \times 10^{-2} \pm 5.60 \times 10^{-3}$	$4.17 \times 10^{-5} \pm 3.00 \times 10^{-5}$	$2.62 \times 10^{-4} \pm 1.29 \times 10^{-4}$
Paper-C [107]	$4.20 \times 10^{-4} \pm 7.15 \times 10^{-6}$	$1.30 \times 10^{-6} \pm 2.20 \times 10^{-8}$	$2.73 \times 10^{-4} \pm 3.48 \times 10^{-4}$	$1.04 \times 10^{-6} \pm 1.44 \times 10^{-6}$	$6.63 \times 10^{-6} \pm 9.33 \times 10^{-6}$
Paper-S [107]	$1.03 \times 10^{-3} \pm 1.16 \times 10^{-5}$	$1.07 \times 10^{-6} \pm 1.05 \times 10^{-8}$	$2.02 \times 10^{-4} \pm 5.94 \times 10^{-6}$	$1.22 \times 10^{-7} \pm 8.82 \times 10^{-10}$	$6.40 \times 10^{-8} \pm 8.30 \times 10^{-10}$
Mean	$1.37 \times 10^{-3} \pm 1.04 \times 10^{-3}$	$3.82 \times 10^{-7} \pm 5.06 \times 10^{-7}$	$8.96 \times 10^{-3} \pm 1.13 \times 10^{-2}$	$1.05 \times 10^{-5} \pm 1.71 \times 10^{-5}$	$4.34 \times 10^{-5} \pm 9.06 \times 10^{-5}$

Table A.2: Complete origin-relative table, presenting both the results and the variance of the data over the 3 runs.

Sequence [#frames]	Raw vertex position		NRIC		
	Chamfer	NMSE	Chamfer	NMSE	NRIC Loss
Low Poly [48]	$2.85 \times 10^{-3} \pm 7.94 \times 10^{-5}$	$5.32 \times 10^{-5} \pm 4.11 \times 10^{-9}$	$4.01 \times 10^{-3} \pm 1.38 \times 10^{-4}$	$2.64 \times 10^{-5} \pm 1.02 \times 10^{-8}$	$5.27 \times 10^{-5} \pm 3.92 \times 10^{-8}$
High Poly [48]	$2.83 \times 10^{-3} \pm 8.35 \times 10^{-5}$	$5.17 \times 10^{-5} \pm 3.54 \times 10^{-9}$	$3.62 \times 10^{-3} \pm 1.25 \times 10^{-4}$	$2.55 \times 10^{-5} \pm 9.79 \times 10^{-9}$	$3.07 \times 10^{-5} \pm 2.15 \times 10^{-8}$
CW-02 [216]	$3.70 \times 10^{-3} \pm 3.60 \times 10^{-4}$	$4.19 \times 10^{-5} \pm 4.42 \times 10^{-8}$	$1.31 \times 10^{-2} \pm 3.11 \times 10^{-3}$	$4.06 \times 10^{-5} \pm 9.25 \times 10^{-6}$	$2.31 \times 10^{-4} \pm 1.01 \times 10^{-6}$
CW-04 [226]	$3.89 \times 10^{-3} \pm 2.87 \times 10^{-4}$	$6.81 \times 10^{-5} \pm 2.00 \times 10^{-7}$	$1.60 \times 10^{-2} \pm 1.75 \times 10^{-3}$	$6.75 \times 10^{-5} \pm 2.76 \times 10^{-6}$	$2.11 \times 10^{-4} \pm 4.35 \times 10^{-6}$
Punch-02 [303]	$4.09 \times 10^{-3} \pm 4.14 \times 10^{-4}$	$1.21 \times 10^{-5} \pm 1.63 \times 10^{-5}$	$4.30 \times 10^{-2} \pm 1.92 \times 10^{-3}$	$1.20 \times 10^{-4} \pm 6.18 \times 10^{-5}$	$2.80 \times 10^{-4} \pm 2.66 \times 10^{-5}$
Paper-C [107]	$7.53 \times 10^{-3} \pm 3.13 \times 10^{-4}$	$2.67 \times 10^{-2} \pm 2.09 \times 10^{-6}$	$3.54 \times 10^{-4} \pm 1.82 \times 10^{-5}$	$4.74 \times 10^{-5} \pm 1.79 \times 10^{-8}$	$4.51 \times 10^{-4} \pm 1.07 \times 10^{-7}$
Paper-S [107]	$4.36 \times 10^{-3} \pm 5.85 \times 10^{-5}$	$2.62 \times 10^{-3} \pm 1.17 \times 10^{-7}$	$3.06 \times 10^{-3} \pm 4.21 \times 10^{-4}$	$2.14 \times 10^{-3} \pm 1.01 \times 10^{-4}$	$1.12 \times 10^{-3} \pm 1.27 \times 10^{-8}$
Mean	$4.04 \times 10^{-3} \pm 1.52 \times 10^{-3}$	$3.69 \times 10^{-3} \pm 9.32 \times 10^{-3}$	$1.45 \times 10^{-2} \pm 1.56 \times 10^{-2}$	$3.15 \times 10^{-4} \pm 7.39 \times 10^{-4}$	$3.32 \times 10^{-4} \pm 3.46 \times 10^{-4}$

Table A.3: Complete median-relative variant table, presenting both the results and the variance of the data over the 3 runs.

Sequence [#frames]	Raw vertex position		NRIC		
	Chamfer	NMSE	Chamfer	NMSE	NRIC Loss
Low Poly [24]	$4.15 \times 10^{-2} \pm 3.78 \times 10^{-3}$	$1.92 \times 10^{-4} \pm 1.65 \times 10^{-5}$	$5.00 \times 10^{-2} \pm 2.31 \times 10^{-2}$	$2.16 \times 10^{-4} \pm 1.80 \times 10^{-4}$	$9.08 \times 10^{-4} \pm 7.50 \times 10^{-4}$
High Poly [24]	$4.47 \times 10^{-2} \pm 2.08 \times 10^{-2}$	$2.07 \times 10^{-4} \pm 1.72 \times 10^{-4}$	$5.79 \times 10^{-2} \pm 6.04 \times 10^{-3}$	$2.85 \times 10^{-4} \pm 2.97 \times 10^{-5}$	$4.56 \times 10^{-4} \pm 1.71 \times 10^{-4}$
CW-02 [108]	$1.31 \times 10^{-2} \pm 6.03 \times 10^{-3}$	$7.51 \times 10^{-6} \pm 5.91 \times 10^{-6}$	$2.23 \times 10^{-1} \pm 1.81 \times 10^{-3}$	$1.57 \times 10^{-3} \pm 5.13 \times 10^{-5}$	$1.85 \times 10^{-2} \pm 1.03 \times 10^{-5}$
CW-04 [113]	$1.02 \times 10^{-2} \pm 1.08 \times 10^{-4}$	$6.43 \times 10^{-6} \pm 7.47 \times 10^{-7}$	$1.61 \times 10^{-1} \pm 3.57 \times 10^{-3}$	$1.34 \times 10^{-3} \pm 6.51 \times 10^{-6}$	$1.30 \times 10^{-2} \pm 4.72 \times 10^{-5}$
Paper-C [53]	$2.43 \times 10^{-2} \pm 2.48 \times 10^{-3}$	$6.60 \times 10^{-3} \pm 1.42 \times 10^{-3}$	$9.00 \times 10^{-3} \pm 3.01 \times 10^{-4}$	$9.58 \times 10^{-4} \pm 7.07 \times 10^{-5}$	$2.77 \times 10^{-3} \pm 6.92 \times 10^{-5}$
Paper-S [53]	$2.20 \times 10^{-2} \pm 3.94 \times 10^{-3}$	$5.01 \times 10^{-4} \pm 1.99 \times 10^{-4}$	$4.31 \times 10^{-2} \pm 7.71 \times 10^{-3}$	$5.80 \times 10^{-3} \pm 1.41 \times 10^{-3}$	$6.35 \times 10^{-3} \pm 1.65 \times 10^{-3}$
Mean	$2.60 \times 10^{-2} \pm 1.43 \times 10^{-2}$	$1.25 \times 10^{-3} \pm 2.63 \times 10^{-3}$	$9.07 \times 10^{-2} \pm 8.27 \times 10^{-2}$	$1.69 \times 10^{-3} \pm 2.08 \times 10^{-3}$	$7.00 \times 10^{-3} \pm 7.30 \times 10^{-3}$

Table A.4: Complete frame interpolation table, presenting both the results and the variance of the data over the 3 runs.

Sequence [#frames]	Conv		NRIC		
	Chamfer	NMSE	Chamfer	NMSE	NRIC Loss
Normal					
Low Poly [48]	$3.06 \times 10^{-2} \pm 8.68 \times 10^{-4}$	$1.64 \times 10^{-4} \pm 5.54 \times 10^{-7}$	$4.10 \times 10^{-2} \pm 1.30 \times 10^{-3}$	$1.31 \times 10^{-4} \pm 3.01 \times 10^{-6}$	$4.67 \times 10^{-4} \pm 2.49 \times 10^{-5}$
CW-02 [216]	$9.95 \times 10^{-3} \pm 8.33 \times 10^{-3}$	$3.05 \times 10^{-6} \pm 3.63 \times 10^{-6}$	$2.49 \times 10^{-2} \pm 2.32 \times 10^{-2}$	$2.69 \times 10^{-5} \pm 3.39 \times 10^{-5}$	$4.94 \times 10^{-5} \pm 6.36 \times 10^{-5}$
Paper-S [107]	$2.50 \times 10^{-3} \pm 7.24 \times 10^{-4}$	$5.03 \times 10^{-6} \pm 4.16 \times 10^{-7}$	$3.18 \times 10^{-3} \pm 5.21 \times 10^{-4}$	$9.59 \times 10^{-5} \pm 7.07 \times 10^{-6}$	$1.22 \times 10^{-4} \pm 4.04 \times 10^{-7}$
Mean	$1.44 \times 10^{-2} \pm 1.46 \times 10^{-2}$	$5.74 \times 10^{-5} \pm 9.24 \times 10^{-5}$	$2.30 \times 10^{-2} \pm 1.90 \times 10^{-2}$	$8.48 \times 10^{-5} \pm 5.32 \times 10^{-5}$	$2.13 \times 10^{-4} \pm 2.23 \times 10^{-4}$
Cumulative					
Low Poly [48]	$1.22 \times 10^{-1} \pm 7.89 \times 10^{-4}$	$3.07 \times 10^{-3} \pm 2.92 \times 10^{-5}$	$6.12 \times 10^{-1} \pm 6.29 \times 10^{-3}$	$2.04 \times 10^{-2} \pm 2.65 \times 10^{-4}$	$3.19 \times 10^{-1} \pm 6.79 \times 10^{-3}$
CW-02 [216]	$1.45 \times 10^{-1} \pm 6.69 \times 10^{-2}$	$1.61 \times 10^{-3} \pm 1.31 \times 10^{-3}$	$8.93 \times 10^{-1} \pm 1.07 \times 10^{-2}$	$1.98 \times 10^{-2} \pm 4.90 \times 10^{-4}$	$1.29 \times 10^{-1} \pm 2.33 \times 10^{-3}$
Paper-S [107]	$9.42 \times 10^{-2} \pm 3.91 \times 10^{-3}$	$1.59 \times 10^{-2} \pm 3.19 \times 10^{-3}$	$2.16 \times 10^{-1} \pm 3.11 \times 10^{-3}$	$5.83 \times 10^{-2} \pm 1.69 \times 10^{-3}$	$4.05 \times 10^{-2} \pm 1.18 \times 10^{-3}$
Mean	$1.20 \times 10^{-1} \pm 2.54 \times 10^{-2}$	$6.84 \times 10^{-3} \pm 7.84 \times 10^{-3}$	$5.74 \times 10^{-1} \pm 3.40 \times 10^{-1}$	$3.28 \times 10^{-2} \pm 2.21 \times 10^{-2}$	$1.63 \times 10^{-1} \pm 1.42 \times 10^{-1}$

Table A.5: Complete feature-to-feature encoding table, presenting both the results and the variance of the data over the 3 runs.