3D-QAE: Fully Quantum Auto-Encoding of 3D Point Clouds

Lakshika Rathi[1,2,†]
lrathi@wisc.edu
Edith Tretschk[2]
edithtretschk.github.io
Christian Theobalt[2]
theobalt@mpi-inf.mpg.de
Rishabh Dabral[2]
rdabral@mpi-inf.mpg.de
Vladislav Golyanik[2]
golyanik@mpi-inf.mpg.de

[1] Indian Institute of Technology Delhi, New Delhi, India
[2] Max Planck Institute for Informatics, SIC, Saarbrücken, Germany

†work done during an internship at MPI

Abstract

Existing methods for learning 3D representations are deep neural networks trained and tested on classical hardware. Quantum machine learning architectures, despite their theoretically predicted advantages in terms of speed and the representational capacity, have so far not been considered for this problem nor for tasks involving 3D data in general. This paper thus introduces the first quantum auto-encoder for 3D point clouds. Our *3D-QAE* approach is *fully quantum*, *i.e.* all its data processing components are designed for quantum hardware. It is trained on collections of 3D point clouds to produce their compressed representations. Along with finding a suitable architecture, the core challenges in designing such a fully quantum model include 3D data normalisation and parameter optimisation, and we propose solutions for both these tasks. Experiments on simulated gate-based quantum hardware demonstrate that our method outperforms simple classical baselines, paving the way for a new research direction in 3D computer vision. The source code is available at https://4dqv.mpi-inf.mpg.de/QAE3D/.

1 Introduction

Bolstered by the wide accessibility of experimental quantum hardware and software Quantum Computing (QC) simulators, the emerging fields of Quantum Computer Vision (QCV) [8, 10, 19, 22, 45, 69, 71] and Quantum Machine Learning (QML) [1, 5, 9, 26, 29, 32, 37, 58, 64] have witnessed a steadily growing number of research works. Most of them are motivated by multiple advantages QC promises over classical computing such as better complexity classes [47], fast convergence, and lower numbers of parameters in machine learning architectures [9]. While most QCV works [10, 21, 57, 72] exploit the quantum annealing paradigm of QC that solves quadratic binary unconstrained optimisation objectives only (which can be too restrictive for many vision problems), this paper uses gate-based

QC supporting a universal set of operations on qubits. We found this flexibility helpful for designing a fully quantum architecture.

Despite this flurry of works, QC and QML algorithms have not yet been applied to 3D point clouds, which are of core interest in 3D computer vision and graphics communities. We start the exploration by looking at auto-encoding 3D point clouds with known correspondences. Our motivation stems from existing works on shape auto-encoders [65, 74] and quantum auto-encoders in different scenarios [52] suggesting that quantum 3D point cloud auto-encoding should be feasible.

While works on classical mesh auto-encoders [12, 51, 65] often deploy sophisticated components like graph convolutions, they also hint at the strength of a very basic design: flattening the mesh coordinates into a vector, encoding it into a bottleneck latent space via a single fully connected layer followed by an activation function, and then decoding it via another single fully connected layer. For example, CoMA [51] focuses on very small latent spaces in order to show an advantage of its architecture over such a simple baseline. Similarly, DEMEA [65] only outperforms such baselines when using small latent spaces. Since practical quantum computing is in its infancy, sophisticated designs are hardly feasible. However, it is possible to design a Quantum Neural Network (QNN) closely resembling a basic classical architecture. This makes mesh auto-encoding particularly promising and well-suited for us as it means that we can start the exploration into quantum 3D scene representations with simple quantum architectures.

Consequently, this paper introduces 3D-QAE, *the first quantum auto-encoder for 3D point sets and investigates its properties*; see Fig. 1. Even though hybrid networks, which combine classical and quantum components, are widespread in the QML literature [3, 4, 43, 56, 61, 66, 70], we find that classical components dominate the useful processing in hybrid architectures, making up for and hiding the shortcomings of the quantum components. This is not surprising, as fully quantum or hybrid architectures are often reported in the literature not to outperform the classical counterparts [11, 28, 35, 39]. Therefore, to better assess the true potential of quantum computing, we focus on a non-hybrid, *purely quantum method*, *e.g.* we do not apply classical non-linearities but, instead, employ a fully quantum non-linearity. Designing and training fully quantum architectures remains very challenging in general. Thus, we need to carefully normalise the data and introduce auxiliary values in order to deal with the input and output restrictions of quantum circuits when using 3D point clouds. We conduct experiments on articulated human poses from the AMASS dataset [41] to evaluate our design choices, in particular the circuit architecture, the non-linearity, and the optimisation

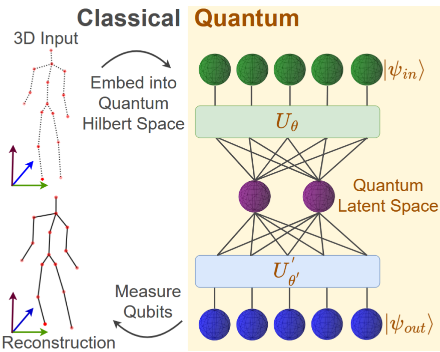


Figure 1: **We use auto-encoding to learn a quantum 3D point cloud representation.** We first embed a classical input into a quantum state $|\psi_{in}\rangle$ of entangled qubits (visualised as spheres), run a learned encoder quantum circuit U_θ on it, and then apply a quantum non-linearity to obtain the quantum latent variable. We then run a learned decoder quantum circuit $U'_{\theta'}$ on the latent vector to obtain the output quantum state $|\psi_{out}\rangle$ and finally measure the qubits to recover a classical, reconstructed output 3D point cloud.

scheme. Summa sumarrum, our primary **technical contributions** are as follows:

- 3D-QAE, a fully quantum gate-based architecture for 3D point clouds auto-encoding,
- Data normalisation scheme to make point sets compatible with quantum circuits, and
- A quantum gate sequence for improved information propagation after the bottleneck.

2 Related Work

Classical Auto-Encoders. Auto-encoders were introduced by Rumelhart and McClelland [53] and have traditionally been used for dimensionality reduction and data analysis [13, 31]. Classical auto-encoders for meshes and point clouds have been used for a variety of applications [6, 12, 20, 25, 62, 63, 68]. A prominent application is learning a latent representation of deformable 3D meshes [67, 68]. Thus, Ranjan *et al.* [51] use spectral graph convolutions [17] with upsampling and downsampling operations to auto-encode meshes of human faces.

Quantum Machine Learning (QML). The hope that quantum systems can outperform classical systems at identifying atypical patterns in data gave rise to the field of QML [9]. The past years have witnessed the birth of quantum analogues of a range of classical machine learning algorithms, including support vector machines [26], principal component analysis [37], and Boltzmann machines [5]. In particular, *Quantum Neural Networks (QNN)* are the quantum analogue of classical neural networks. A QNN consists of parametrised quantum circuits applied on an input quantum state generated by a feature map [32]. Sometimes, QNNs achieve [1] higher expressibility and better trainability than classical counterparts.

Quantum Auto-Encoders (QAE) are a special case of QNNs. QAE often follow layered architecture design, which allows to use quantum systems in tasks analogous to classical machine learning. They are often designed as hybrid systems combining quantum and classical layers to cater to a range of tasks: labeling classical data [42], reconstructing and sampling of biological drugs [36], searching anomalous data points in a given classical dataset [54], and compressing information to enable efficient communication between a client and a server in a quantum cloud computing setting [75]. Similarly to how classical auto-encoders learn low-dimensional representations of classical data, quantum auto-encoders [52, 61] can be used to represent quantum data in a compressed form: Bravo-Prieto [14] demonstrates the use of parametrised quantum circuits as variational models to compress Ising models and handwritten digits with high fidelity. Effective error detection is crucial for contemporary quantum devices and quantum auto-encoders can be efficiently used for detecting and mitigating quantum errors [73] [38]. Finally, they have also been shown to effectively denoise Greenberger-Horne-Zeilinger states from noisy quantum channels [2].

Quantum Computer Vision and Graphics (QCV/CG). A range of quantum methods have been developed for tasks in computer vision like object detection [27] and image classification [4, 15]. The limited number of physical qubits in gate-based models has led to a preference for adiabatic quantum computing for most tasks in computer vision [34], such as robust fitting [19, 21], multiple object tracking [48], permutation synchronisation [10, 71], and transformation estimation [22, 45, 58]. However, hybrid quantum-classical gate-based models are also emerging [69]. Our method is designed for the gate-based quantum computing paradigm and it is the first to learn 3D point cloud representations with QML.

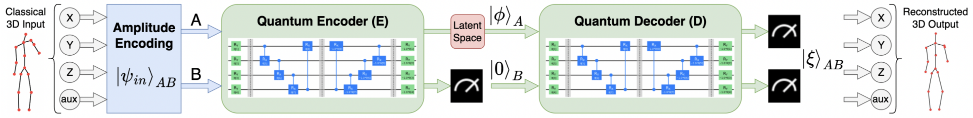


Figure 2: **3D-QAE, a quantum point cloud auto-encoder.** We prepare a classical 3D point cloud as input and then encode it into a quantum state vector $|\psi_{in}\rangle$ of two sets of qubits, A and B, via amplitude encoding. The encoder E (visualised here with $J{=}1$ block) acts on this state vector via a learned unitary transform implemented by a parametrised quantum circuit. At the bottleneck, we remove the information stored in the qubits B. This removal acts as a quantum non-linearity whose output is the latent vector $|\phi\rangle$ of qubits A. We re-initialise qubits B to $|0\rangle$ and let the decoder D, whose architecture is the same as E's, transform qubits A and B. We then measure the output of D to obtain the state vector $|\xi\rangle$, which we can classically process in a loss function or convert to the final 3D output reconstruction.

3 Method

We propose *3D-QAE*, *i.e.* a fully quantum, circuit-based auto-encoder for registered 3D point clouds. When using current classical hardware to simulate quantum hardware, it scales to dozens of points per point cloud. Fig. 2 shows the scheme of 3D-QAE. For background on gate-based quantum computing, we refer to the supplementary material.

Like classical auto-encoders, quantum auto-encoders consist of an encoder and a decoder, with a bottleneck latent space in the middle. First, the classical data is encoded into a quantum state (Sec. 3.1) that is passed to the encoder (Sec. 3.2). The encoder output is compressed into a low-dimensional latent space, before being passed to the decoder (Sec. 3.3). Finally, we optimise for the best parameters of the auto-encoder by encouraging similarity between the decoder output and the input (Sec. 3.4).

3.1 Input Data Normalisation and Encoding

We take as input a classical 3D point cloud $\{\mathbf{v}_i \in \mathbb{R}^3\}_{i=0}^{V-1}$ with V vertices. Several steps are necessary to turn this point cloud into a *state vector* that can be input into a quantum circuit. Furthermore, as we discuss later in Sec. 3.4, the output of the auto-encoder necessarily lies in the positive octant since it is a vector $(a_0, \dots, a_{2^N-1})$ with $a_i \geq 0$ and $\sum_i |a_i|^2 = 1$.

Data Normalisation. Since the output of the auto-encoder always lies in the positive octant, we also normalise our data to lie in the positive octant. We first compute a tight, cubic bounding box around all vertices in the dataset. We then normalise our data by shifting and isotropically rescaling it such that the bounding box coincides with the unit cube in the positive octant. See our supplement for further details.

Encoding the Classical Data. Our inputs contain dozens of vertices, which is substantial for current quantum systems. We

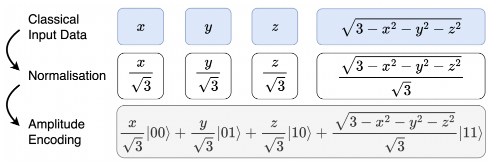


Figure 3: Data Encoding ($V{=}1$). We encode a 3D vertex by adding an auxiliary value, normalising it to a norm 1.0 and turning it into a quantum state vector via amplitude encoding.

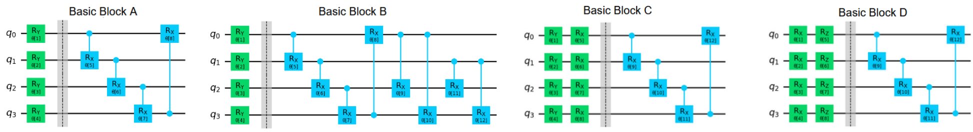


Figure 4: Different quantum blocks. “A” and “D” are inspired by Sim *et al.* [60].

use amplitude encoding to transform our classical data into a quantum state vector, since it allows us to utilise the exponentially large Hilbert space; see Fig. 3. Recall that amplitude encoding demands a state vector with the unit norm. Hence, we introduce an auxiliary value $\sqrt{3-x_i^2-y_i^2-z_i^2}$ for each normalised vertex $\tilde{\mathbf{v}}_\mathbf{i}=(x_i,y_i,z_i)$ as a constant normalisation factor across all point clouds. Moreover, the state vector size is always a power of 2 and we fill up the remaining entries with zeros to get a state vector of size 2^N. Finally, the norm of the state vector is one and we, thus, need to normalise all values by $\sqrt{3V}$:

$$|\psi_{in}\rangle = \frac{1}{\sqrt{3V}}\sum_{i=0}^{V-1}\left(x_i\,|3i\rangle + y_i\,|3i+1\rangle + z_i\,|3i+2\rangle + \sqrt{3-x_i^2-y_i^2-z_i^2}\,|3V+i\rangle\right) + \sum_{j=4V}^{2^N-1} 0\,|j\rangle\,. \quad (1)$$

3.2 Quantum Circuit Design

We next describe the design of our architecture with quantum circuits. The *basic block* of such a quantum circuit is a layer of rotation gates L_R followed by a layer L_{CR} of controlled rotation gates. Naturally, there are several possible combinations of these gates. Thus, we investigate different architectures for these basic blocks (Fig. 4). We start with the simple architecture of the basic block “A”, which uses *linear* entanglement where each qubit is entangled with two neighbours in a circular fashion. We experimented with different combinations of gate types and found R_Y gates in the L_R layer and CR_X gates in the L_{CR} layer to work best. This is consistent with prior work that measures the expressibility and entangling capacity of circuits under different combinations of these rotation and controlled rotation gates [60]. The design of “B” is a variant of “A” that additionally considers the bottleneck in its design. Specifically, we add entangling gates between qubits that get removed at the bottleneck and those that remain (as we will discuss later in Sec. 3.3), enabling better information propagation. Finally, the blocks “C” and “D” are alternative ways of increasing the capacity of the block “A” without adding entangling gates.

We combine two basic blocks (or their inverses), which we call F and S, into a *block* $X=SF$. We investigate two types: (1) the “repeat” type uses the same architecture for both basic blocks, while (2) the “inverse” type uses the inverse architecture of F for S; see supplement C for a visualisation. We initialise the parameters of F randomly. For S, we investigate two initialisation schemes: (1) the “random” scheme initialises it with random parameters, while (2) the “identity” scheme depends on the type of the block. In the second case, for the “repeat” type, we use the same initial parameters as for F, and for the “inverse” type, we use the initial parameters that make S the inverse of F. The “inverse” type with the “identity” initialisation is also known as *identity-block initialisation* [23] that is meant to prevent the model from being stuck in a barren plateau at the beginning of training.

Finally, we build a quantum circuit by chaining J blocks together: $U=X_{J-1}\cdots X_0$. We always use the same architecture scheme and initialisation scheme for all X_j in a circuit U.

3.3 Full Architecture

We next employ any such design of a single quantum circuit (Sec. 3.2) to build the overall architecture of our quantum auto-encoder, whose parameters are denoted by θ; see Fig. 2.

Specifically, both the encoder and decoder each consist of a single quantum circuit with the same number of blocks J, mirroring the classical fully connected design.

Encoder. The encoder reduces the dimension of its input, which is non-trivial to implement when using a quantum circuit because it is a unitary transform. To that end, we define the encoder in terms of two subsystems of qubits, namely A with N_A qubits and B with $N_B{=}N{-}N_A$ qubits. A contains the information ultimately passed on to the decoder and B is discarded at the end of the encoder. The input state vector $|\psi_{in}\rangle_{AB} = |\psi_{in}\rangle$ is the state vector of all qubits in A and B and the encoder E first acts on it as $|\phi\rangle_{AB} = E\,|\psi_{in}\rangle_{AB}$.

Bottleneck. To map into the latent space, we discard the information present in the qubit subsystem B by *tracing out* its qubits. This is analogous to marginalising out N_B binary dimensions of an N-dimensional probability distribution if we treat the *squared* amplitudes of $|\phi\rangle_{AB}$ as a distribution. This partial trace introduces a quantum non-linearity into the otherwise unitary auto-encoder. This trace sums over all basis states $|i\rangle_B \in (\mathbb{C}^2)^{\otimes N_B}$ of B:

$$\rho = \mathrm{Tr}_B[|\phi\rangle_{AB}] = \sum_i \underbrace{(I_A \otimes \langle i|_B)}_{2^{N_A}\times 2^N} \underbrace{(|\phi\rangle_{AB}\langle\phi|_{AB})}_{2^N\times 2^N} \underbrace{(I_A \otimes |i\rangle_B)}_{2^N\times 2^{N_A}}, \tag{2}$$

where $|\phi\rangle_A \in (\mathbb{C}^2)^{\otimes N_A}$, $\langle\phi| = |\phi\rangle^\dagger$ is a Hermitian conjugate and, hence, a row vector, and I_A is the $2^{N_A}\times 2^{N_A}$ identity. We treat $I_A \otimes \langle i|_B$, which is a $2^{N_A}\times 2^{N_A}\times 2^{N_B}$ tensor, as its flattened $2^{N_A}\times 2^{N_A}2^{N_B}$ version. On its diagonal, the $2^{N_A}\times 2^{N_A}$ matrix ρ contains the squared amplitudes of $|\phi\rangle_A$. We use amplitude encoding to turn these amplitudes into the state $|\phi\rangle_A$, which is the latent code of the auto-encoder, *i.e.* a learned quantum 3D point cloud representation.

Decoder. We face an analogous problem with the decoder D as with the encoder: We need to map from a lower-dimensional space to a higher-dimensional one, using a unitary transform. We achieve this by expanding $|\phi\rangle_A$ with qubits of B, which are freshly initialised to $|0\rangle_B$. We then obtain $|\xi\rangle = |\xi\rangle_{AB} = D(|\phi\rangle_A \otimes |0\rangle_B)$.

3.4 Training

To determine the best set of parameters θ, we need to define a loss function and then update the parameters accordingly. However, *measuring* $|\xi\rangle$ yields a single N-bit output string, which contains little information and is not differentiable, preventing effective, gradient-based optimisation. We circumvent this by running the auto-encoder multiple times. This yields an empirical estimate of the frequency of each bit string, *i.e.* of the probability of each dimension of the state vector; in simulation, a single run of the auto-encoder is sufficient to obtain all probabilities. We treat these probability amplitudes $(\alpha_0,\ldots,\alpha_{2^N-1})$ as the output and next process them in a classical loss function. We treat $(\alpha_0,\alpha_1,\ldots,\alpha_{3V-1})$ as the predicted vertices and $(\alpha_{3V},\ldots,\alpha_{4V-1})$ as the predicted auxiliary values. We then encourage these output amplitudes of $|\xi\rangle$ to be similar to the input amplitudes of $|\phi_{in}\rangle$ with the loss $\mathcal{L}_{rec}$:

$$\mathcal{L}_{rec} = \sum_{i=0}^{V-1}\left(\sqrt{\zeta_i} + \left|\alpha_{3V+i} - \frac{\sqrt{3-x_i^2-y_i^2-z_i^2}}{\sqrt{3V}}\right|\right) + \sum_{j=4V}^{2^N-1}|\alpha_j|, \text{with } \zeta_i = \left(\alpha_{3i} - \frac{x_i}{\sqrt{3V}}\right)^2 + \left(\alpha_{3i+1} - \frac{y_i}{\sqrt{3V}}\right)^2 + \left(\alpha_{3i+2} - \frac{z_i}{\sqrt{3V}}\right)^2. \tag{3}$$

To minimise $\mathcal{L}_{rec}$, we use the Adam optimiser [30] for backpropagation with an initial learning rate of 10^{-2} and beta values of 0.9 and 0.99. We apply a loss-based learning rate schedule, use a batch size of 1, and train for $10k$ epochs. We use the Pytorch [49] interface in PennyLane [7] for noise-free quantum simulation on the CPU.

4 Experimental Evaluation

We next evaluate the performance of our method qualitatively and quantitatively. We use the AMASS dataset [41] with 3D human motion capture data (joint locations) for various poses with a high variety of articulations. We thus quantify the reconstruction error via the mean Euclidean distance in *cm* across all joints. We temporally downsample the data to 12 fps and select the first $10k$ poses in temporal order. The resulting data is split into training and test data with a 80:20 split. This corresponds to a training motion of 16 *sec* and a test motion of 4 *sec* in each chunk. We use 16 joints from the SMPL model [40] and normalise out the global transform. We encode human poses with $V{=}16$ vertices into state vectors of length 2^6, corresponding to a six-qubit circuit. At the bottleneck, we remove two qubits, which yields a latent code of length 16. Unless stated otherwise, we use $J{=}8$ blocks with block "B" in the *repeat* architecture with the *identity* initialisation, as we found these to work best.

4.1 Comparisons

Fig. 5 shows qualitative results using the proposed method. The reconstructed poses follow the ground-truth states well, with some occasional coarse details.

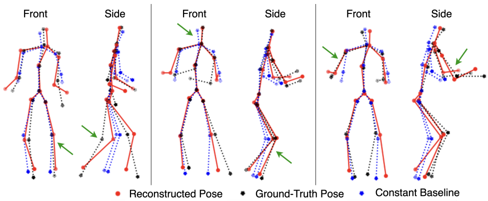


Figure 5: Qualitative results of our fully quantum architecture.

Baselines. We compare to three classical (non-quantum) baselines. First, independent of the input, the *constant baseline* always predicts the mean mesh, *i.e.* the average of all the meshes in the training data. Second, as described in Sec. 1, a basic classical architecture using fully connected layers as the encoder and decoder is quite powerful. We define such a *classical fully connected baseline* by replacing the quantum encoder and decoder with a fully connected layer each in our architecture. We use an ELU non-linearity [16] at the bottleneck and match the number of parameters in the fully connected layers with those in the quantum architecture. We do not use auxiliaries and do not fill up the input vector with zeros. Third, to better locate performance bottlenecks, we also experiment with a *classical mimic baseline* whose design sticks very closely to the quantum architecture. The *only* change we make to our architecture is replacing quantum circuits with square fully connected layers whose output we normalise to norm 1.

Quantitative Results. Quantitative results are reported in Tab. 1. We find that 3D-QAE outperforms the constant baseline but is barely competitive with the best classical baseline. Note that the fully connected baseline matched to four blocks achieves a mean Euclidean distance of 11.51 *cm*. In addition, the mimic baseline performs close to the fully connected baseline, which indicates that the design choices in which they differ are, most probably, not the limiting factor for the performance of our method.

Method	mean Euclidean distance
Constant baseline	13.58
Classical mimic baseline	3.75
Fully connected baseline (8 blocks)	8.37
Fully connected baseline (16 blocks)	3.85
Ours (8 blocks)	10.86
Ours (16 blocks)	10.45

Table 1: Comparisons against classical baselines. We match the number of parameters in the classical fully connected baseline to those of our architecture with 8 or 16 blocks.

4.2 Analysis

Circuit Design. We investigate the performance along three axes: (1) which basic block to use, (2) *repeat vs inverse* design, and (3) *random vs identity* initialisation. As is common in the QML literature, we first compare the different architectures using the same number of blocks, namely J=8; see Tab. 2. We see that basic block "B" with the *repeat* design and *identity* initialisation performs the best. In general, barren plateaus often arise with generic, hardware-efficient circuits like ours. However, we find the *identity* initialisation (*i.e.* inverse architecture with identity initialisation) that mitigates them yields no advantage in our case. Thus, barren plateaus appear to not be a limiting factor for the current best-performing circuit.

Architecture	**Initialisation**	**Basic Block**			
		"A"	"B"	"C"	"D"
Repeat	Random	12.86	12.58	12.16	12.52
	Identity	11.36	**10.86**	12.03	11.52
Inverse	Random	13.39	11.94	12.56	11.82
	Identity	12.26	11.45	12.41	12.11

Table 2: M. Euclidean distance for different circuit designs for J=8 blocks in the encoder and decoder.

However, the number of quantum gates (and, in turn, parameters) differs across the different block types. We, thus, compare the different block types by matching the number of parameters to those of "B". To this end, we increase the number of blocks for the "A", "C", and "D" types accordingly. The results are in Tab. 3. While those circuits using basic blocks other than "B" improve their performance, "B" still shows the best performance overall. This validates our choice to add entangling gates between qubits that were removed and qubits that remained at the bottleneck.

Architecture	**Initialisation**	**Basic Block**			
		"A"	"B"	"C"	"D"
Repeat	Random	11.52	12.58	11.73	11.12
	Identity	11.37	**10.86**	11.74	11.95
Inverse	Random	11.99	11.94	12.34	12.33
	Identity	12.38	11.45	11.87	11.92

Table 3: M. Euclidean distance for different circuit designs. We use J=8 blocks for basic block "B". For the other basic blocks, we use that J that matches the number of parameters of the basic block "B" architecture the closest.

Number of Blocks. Like classical neural architectures, quantum circuits become more expressive (and harder to implement and optimise) the deeper they are. Here, we take a closer look at how the quality and the quantitative error change with the number of blocks J. Fig. 7(a) contains qualitative results that show a clear improvement with more blocks. Fig. 6(a) confirms this quantitatively. Using J=16 rather than 8 blocks improves results somewhat. However, this comes at an impracticable training time, as it doubles the 45 hours it takes on the CPU for J=8.

Number of Bottleneck Qubits. We discard several qubits at the bottleneck and thereby reduce the amount of information transmitted into the decoder. Figs. 6(b) and 7(b) show

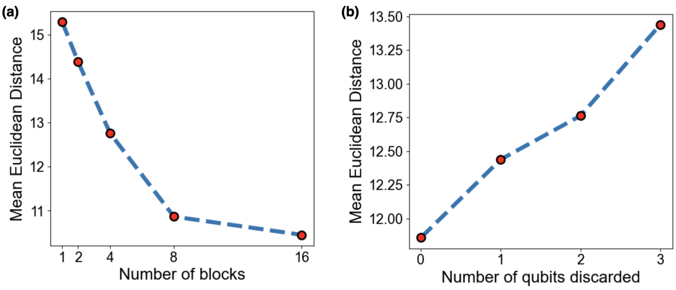
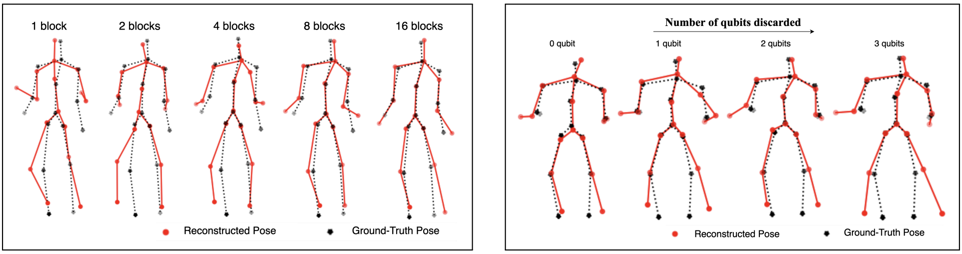


Figure 6: Mean Euclidean distance for a varying number of (a) blocks and (b) qubits discarded in the bottleneck.



(a) (b)

Figure 7: Examples for a varying numbers of (a) blocks J and (b) bottelneck qubits.

numerical and qualitative results, respectively. As expected, a smaller latent space leads to worse reconstruction accuracy. We note that, in contrast to the classical setting, every discarded qubit reduces the amount of information by a factor of two instead of linearly.

Non-Linearity. Tab. 4 shows that 3D-QAE with a quantum non-linearity (QNL) is on par with a carefully designed classical activation (ELU [16]). This is supported by the mimic baseline (with QNL) that is on par with the fully connected baseline (with ELU); see Tab. 1 for numerical comparisons.

Number of Blocks	ELU	QE-CD	CE-QD	Ours
4	12.97	7.06	11.25	12.76
8	10.98	6.34	10.20	10.86

Table 4: Mean Euclidean distances. Instead of marginalisation, *ELU* takes a sub-vector from the encoder and applies an ELU at the bottleneck. *QE-CD* is a hybrid with a quantum encoder and a classical decoder, *CE-QD* uses a classical encoder and a quantum decoder.

Hybrid Models. We next shortly look at hybrid models. We replace either only the encoder or only the decoder with a classical fully connected layer as in the mimic baseline. Tab. 4 shows that replacing quantum parts with classical ones improves the performance.

Simpler Tasks. Finally, we explore simpler auto-encoding tasks to better understand the gap between the constant baseline and our method. To this end, we look at three different body parts with three vertices each, rather than the entire body. Tab. 5 shows that our method outperforms the constant baseline on these easier tasks even more than on the full body. This suggests that task difficulty (*e.g.*, motion complexity) substantially influences the performance of our method. The training time decreases exponentially with the number of qubits. For these simpler experiments in the $J = 8$ case, it requires 9 hours on the CPU.

Method	Left Leg	Right Arm	Spine	Full Body
Constant Baseline	10.7	22.1	7.4	13.6
Ours (8 blocks)	**3.2**	**6.5**	**1.5**	**10.9**
Ratio (Ours / Con.)	0.30	0.29	0.20	0.80

Table 5: Auto-encoding body parts. Mean Euclidean distances are in *cm*.

4.3 Discussion

Since quantum supremacy has not yet been achieved in practical settings, it was to be expected that our method would not outperform classical baselines. We have eliminated a number of possible reasons for the gap between our best-performing fully quantum architecture and the classical baselines. The performance of the classical mimic baseline suggests that the limiting factor lies with the quantum circuit itself. However, as discussed, barren plateaus appear to not be an issue, as mitigating them has no discernible effect on performance. Furthermore, basic block "B" designed with the bottleneck in mind performs slightly better than the alternatives. Still, we found only small performance differences over a wide range of typical hardware-efficient circuit designs. The hybrid models indicate that enough information remains at the bottleneck when using a quantum encoder. The issue thus rather lies with the expressiveness of the quantum decoder (perhaps because it is unitary and, thus, represents an *orthonormal* set of basis shapes for the output). However, while further increasing the number of blocks does improve the accuracy, the returns diminish quickly. Even larger architectures are infeasible in reasonable runtime with current software and hardware.

Future Work. Many quantum papers [24, 33, 44, 59, 69] that investigate vision tasks focus on classification rather than regression (*e.g.* digit classification on MNIST [18]). Future work on applying quantum computing to 3D tasks could thus also address classification problems. In a narrower sense, work on equivariant quantum networks [46, 50, 55] that explicitly accounts for global translation and rotation are an interesting future avenue.

5 Conclusion

This paper investigates 3D point cloud auto-encoding with a fully quantum architecture. Our data normalisation scheme with amplitude encoding is crucial and allows representing classical data (3D point clouds) as inputs for our approach. We explore a wide range of design choices and succeed in overcoming a surprisingly challenging hurdle, *i.e.* outperforming a constant baseline. We observe in the experiments that replacing either the encoder or decoder with their classical counterparts decreases the reconstruction error. Moreover, the simpler the motions, the lower the error our fully quantum method achieves. Our results suggest that a substantial gap remains between classical and quantum architectures, which is in line with prior work in QML. This first study and our thorough analysis plausibly eliminate many reasons for this gap, and future work could further attempt to reduce and eventually close it.

Acknowledgement. This work was supported by the ERC consolidator grant *4DReply* (770784).

References

[1] Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 2021.

[2] Tom Achache, Lior Horesh, and John Smolin. Denoising quantum states with Quantum Autoencoders – Theory and Applications. *Conference on Quantum Information Processing (QIP)*, 2021.

[3] Mahabubul Alam and Swaroop Ghosh. Deepqmlp: A scalable quantum-classical hybrid deep neural network architecture for classification. In *International Conference on VLSI Design and International Conference on Embedded Systems (VLSID)*, 2022.

[4] Mahabubul Alam, Satwik Kundu, Rasit Onur Topaloglu, and Swaroop Ghosh. Quantum-classical hybrid machine learning for image classification. In *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2021.

[5] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum boltzmann machine. *Physical Review X*, 2018.

[6] Timur Bagautdinov, Chenglei Wu, Jason Saragih, Pascal Fua, and Yaser Sheikh. Modeling facial geometry using compositional vaes. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] Ville Bergholm et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv e-prints*, 2018.

[8] Harshil Bhatia, Edith Tretschk, Zorah Lähner, Marcel Benkner, Michael Moeller, Christian Theobalt, and Vladislav Golyanik. Ccuantumm: Cycle-consistent quantum-hybrid matching of multiple shapes. In *Computer Vision and Pattern Recognition (CVPR)*, 2023.

[9] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 2017.

[10] Tolga Birdal, Vladislav Golyanik, Christian Theobalt, and Leonidas Guibas. Quantum permutation synchronization. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] Denis Bokhan, Alena S. Mastiukova, Aleksey S. Boev, Dmitrii N. Trubnikov, and Aleksey K. Fedorov. Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning. *Front. in Phys.*, 10:1069985, 2022.

[12] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *International Conference on Computer Vision (ICCV)*, 2019.

[13] Herve Bourlard and Y Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 1988.

[14] Carlos Bravo-Prieto. Quantum autoencoders with enhanced data encoding. *Machine Learning: Science and Technology*, 2021.

[15] Avinash Chalumuri, Raghavendra Kune, S. Kannan, and B. S. Manoj. Quantum-enhanced deep neural network architecture for image scene classification. *Quantum Information Processing*, 2021.

[16] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *International Conference on Learning Representations (ICLR)*, 2016.

[17] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[18] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 2012.

[19] Anh-Dzung Doan, Michele Sasdelli, David Suter, and Tat-Jun Chin. A hybrid quantum-classical algorithm for robust fitting. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[20] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] Matteo Farina, Luca Magri, Willi Menapace, Elisa Ricci, Vladislav Golyanik, and Federica Arrigoni. Quantum multi-model fitting. In *Computer Vision and Pattern Recognition (CVPR)*, 2023.

[22] Vladislav Golyanik and Christian Theobalt. A quantum computational approach to correspondence problems on point sets. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[23] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*, 2019.

[24] Daniel Guijo, Victor Onofre, Gianni Del Bimbo, Samuel Mugel, Daniel Estepa, Xabier De Carlos, Ana Adell, Aizea Lojo, Josu Bilbao, and Roman Orus. Quantum artificial vision for defect detection in manufacturing. *arXiv e-prints*, 2022.

[25] Sara Hahner and Jochen Garcke. Mesh convolutional autoencoder for semi-regular meshes of different sizes. In *Winter Conference on Applications of Computer Vision (WACV)*, 2022.

[26] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 2019.

[27] Ling Hu and Qiang Ni. Quantum automated object detection algorithm. In *International Conference on Automation and Computing (ICAC)*, 2019.

[28] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Information-theoretic bounds on quantum advantage in machine learning. *Phys. Rev. Lett.*, 126:190505, 2021.

[29] Sofiene Jerbi, Lukas J. Fiderer, Hendrik Poulsen Nautrup, Jonas M. Kübler, Hans J. Briegel, and Vedran Dunjko. Quantum machine learning beyond kernel methods. *Nature Communications*, 2023.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

[31] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *Aiche Journal*, 1991.

[32] Yunseok Kwak, Won Joon Yun, Soyi Jung, and Joongheon Kim. Quantum neural networks: Concepts, applications, and challenges. In *International Conference on Ubiquitous and Future Networks (ICUFN)*, 2021.

[33] Jonas Landman, Natansh Mathur, Yun Yvonna Li, Martin Strahm, Skander Kazdaghli, Anupam Prakash, and Iordanis Kerenidis. Quantum methods for neural networks and application to medical image classification. *Quantum*, 2022.

[34] Harashta Tatimma Larasati, Thi-Thu-Huong Le, and Howon Kim. Trends of quantum computing applications to computer vision. In *International Conference on Platform Technology and Service (PlatCon)*, 2022.

[35] Bayo Lau, Prashant S Emani, Jackson Chapman, Lijing Yao, Tarsus Lam, Paul Merrill, Jonathan Warrell, Mark B Gerstein, and Hugo Y K Lam. Insights from incorporating quantum computing into drug design workflows. *Bioinformatics*, 39(1), 12 2022.

[36] Junde Li and Swaroop Ghosh. Scalable variational quantum circuits for autoencoder-based drug discovery. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022.

[37] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 2014.

[38] David F Locher, Lorenzo Cardarelli, and Markus Müller. Quantum error correction with quantum autoencoders. *Quantum*, 2023.

[39] Owen Lockwood and Mei Si. Playing atari with hybrid quantum-classical reinforcement learning. In *NeurIPS Workshop on Pre-registration in Machine Learning*, 2020.

[40] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 2015.

[41] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision (ICCV)*, 2019.

[42] Stefano Mangini, Alessia Marruzzo, Marco Piantanida, Dario Gerace, Daniele Bajoni, and Chiara Macchiavello. Quantum neural network autoencoder and classifier applied to an industrial case study. *Quantum Machine Intelligence*, 2022.

[43] Andrea Mari, Thomas R. Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. Transfer learning in hybrid classical-quantum neural networks. *Quantum*, 2020.

[44] Natansh Mathur, Jonas Landman, Yun Yvonna Li, Martin Strahm, Skander Kazdaghli, Anupam Prakash, and Iordanis Kerenidis. Medical image classification via quantum neural networks. *arXiv e-prints*, 2021.

[45] Natacha Kuete Meli, Florian Mannel, and Jan Lellmann. An iterative quantum approach for transformation estimation from point sets. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[46] Quynh T. Nguyen, Louis Schatzki, Paolo Braccia, Michael Ragone, Patrick J. Coles, Frederic Sauvage, Martin Larocca, and M. Cerezo. Theory for equivariant quantum neural networks. *arXiv e-prints*, 2022.

[47] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. 2000.

[48] Yesul Park, L. Minh Dang, Sujin Lee, Dongil Han, and Hyeonjoon Moon. Multiple object tracking in deep learning approaches: A survey. *Electronics*, 2021.

[49] Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[50] Michael Ragone, Paolo Braccia, Quynh T. Nguyen, Louis Schatzki, Patrick J. Coles, Frederic Sauvage, Martin Larocca, and M. Cerezo. Representation theory for geometric quantum machine learning. *arXiv e-prints*, 2022.

[51] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, 2018.

[52] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2017.

[53] David E. Rumelhart and James L. McClelland. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, 1987.

[54] Alona Sakhnenko, Corey O'Meara, Kumar JB Ghosh, Christian B Mendl, Giorgio Cortiana, and Juan Bernabé-Moreno. Hybrid classical-quantum autoencoder for anomaly detection. *Quantum Machine Intelligence*, 2022.

[55] Louis Schatzki, Martin Larocca, Quynh T. Nguyen, Frederic Sauvage, and M. Cerezo. Theoretical guarantees for permutation-equivariant quantum neural networks. *arXiv e-prints*, 2022.

[56] Alessandro Sebastianelli, Daniela Alessandra Zaidenberg, Dario Spiller, Bertrand Le Saux, and Silvia Liberata Ullo. On circuit-based hybrid quantum neural networks for remote sensing imagery classification. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2021.

[57] Marcel Seelbach Benkner, Zorah Lähner, Vladislav Golyanik, Christof Wunderlich, Christian Theobalt, and Michael Moeller. Q-match: Iterative shape matching via quantum annealing. In *International Conference on Computer Vision (ICCV)*, 2021.

[58] Marcel Seelbach Benkner, Maximilian Krahn, Edith Tretschk, Zorah Lähner, Michael Moeller, and Vladislav Golyanik. QuAnt: Quantum Annealing with Learnt Couplings. *International Conference on Learning Representations (ICLR)*, 2023.

[59] Ruoxi Shi, Hao Tang, and Xian min Jin. Training a quantum pointnet with nesterov accelerated gradient estimation by projection. In *Advances in Neural Information Processing Systems (NeurIPS) Workshops*, 2020.

[60] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2019.

[61] Maiyuren Srikumar, Charles D Hill, and Lloyd CL Hollenberg. Clustering and enhanced classification using a hybrid quantum autoencoder. *Quantum Science and Technology*, 2021.

[62] Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. Variational autoencoders for deforming 3d mesh models. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[63] Qingyang Tan, Lin Gao, Yu-Kun Lai, Jie Yang, and Shihong Xia. Mesh-based autoencoders for localized deformation component analysis. In *AAAI Conference on Artificial Intelligence*, 2018.

[64] Jinkai Tian et al. Recent advances for quantum neural networks in generative learning. *arXiv e-prints*, 2022.

[65] Edgar Tretschk, Ayush Tewari, Michael Zollhöfer, Vladislav Golyanik, and Christian Theobalt. DEMEA: Deep Mesh Autoencoders for Non-Rigidly Deforming Objects. *European Conference on Computer Vision (ECCV)*, 2020.

[66] Maida Wang, Anqi Huang, Yong Liu, Xuming Yi, Junjie Wu, and Siqi Wang. A quantum-classical hybrid solution for deep anomaly detection. *Entropy*, 2023.

[67] Siming Yan, Zhenpei Yang, Haoxiang Li, Li Guan, Hao Kang, Gang Hua, and Qixing Huang. Implicit autoencoder for point cloud self-supervised representation learning. *arXiv preprint arXiv:2201.00785*, 2022.

[68] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud autoencoder via deep grid deformation. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[69] Yuan-Fu Yang and Min Sun. Semiconductor defect detection by hybrid classical-quantum deep learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[70] Samuel Yen-Chi Chen, Tzu-Chieh Wei, Chao Zhang, Haiwang Yu, and Shinjae Yoo. Hybrid Quantum-Classical Graph Convolutional Network. *arXiv e-prints*, 2021.

[71] Alp Yurtsever, Tolga Birdal, and Vladislav Golyanik. Q-fw: A hybrid classical-quantum frank-wolfe for quadratic binary optimization. In *European Conference on Computer Vision (ECCV)*, 2022.

[72] Jan-Nico Zaech, Alexander Liniger, Martin Danelljan, Dengxin Dai, and Luc Van Gool. Adiabatic quantum computing for multi object tracking. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[73] Xiao-Ming Zhang, Weicheng Kong, Muhammad Usman Farooq, Man-Hong Yung, Guoping Guo, and Xin Wang. Generic detection-based error mitigation using quantum autoencoders. *Physical Review A*, 2021.

[74] Yi Zhou, Chenglei Wu, Zimo Li, Chen Cao, Yuting Ye, Jason Saragih, Hao Li, and Yaser Sheikh. Fully convolutional mesh autoencoder using efficient spatially varying kernels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[75] Yan Zhu, Ge Bai, Yuexuan Wang, Tongyang Li, and Giulio Chiribella. Quantum autoencoders for communication-efficient quantum cloud computing. *arXiv e-prints*, 2021.