Quantum Machine Intelligence (2025) 7:103
https://doi.org/10.1007/s42484-025-00327-8

RESEARCH ARTICLE

Hybrid quantum tensor networks for aeroelastic applications

M. Lautaro Hickmann[1] · Pedro Alves[1] · David Quero[2] · Friedhelm Schwenker[3] · Hans-Martin Rieser[1]

Received: 30 July 2025 / Accepted: 17 October 2025

Abstract

We investigate the application of hybrid quantum tensor networks to aeroelastic problems, harnessing the power of Quantum Machine Learning (QML). By combining tensor networks with variational quantum circuits, we demonstrate the potential of QML to tackle complex time series classification and regression tasks. Our results showcase the ability of hybrid quantum tensor networks to achieve high accuracy in binary classification. Furthermore, we observe promising performance in regressing discrete variables. While hyperparameter selection remains a challenge, requiring careful optimisation to unlock the full potential of these models, this work contributes significantly to the development of QML for solving intricate problems in aeroelasticity. We present an end-to-end trainable hybrid algorithm. We first encode time series into tensor networks to then utilise trainable tensor networks for dimensionality reduction, and convert the resulting tensor to a quantum circuit in the encoding step. Then, a tensor network inspired trainable variational quantum circuit is applied to solve either a classification or a multivariate or univariate regression task in the aeroelasticity domain.

Keywords Tensor networks · Quantum machine learning · Hybrid machine learning · Variational quantum circuits

1 Introduction

Simulations of aeroelastic phenomena involve modelling complex fluid dynamics and the structural behaviour of components. For modern aircraft design, a detailed level of fidelity in the modelling of complex aeroelastic phenomena is essential. Increasing modelling fidelity leads to the need of resolving ever finer grids leading to an enormous computational effort for the numerical simulations. Therefore, finding efficient implementations is a key research field in aeroelastics. In particular, this involves developing techniques for the reduced order modelling of nonlinear aerodynamics. These techniques need to consider the complex nonlinear behaviour originated by the compressible, viscous and turbulent flow phenomena while not needing to simulate this behaviour on each grid point. Inherent difficulties are the nonlinear dependence and the high-dimensionality regarding the phase space on the grid required in order to describe such features.

Data driven implementations using machine-learning algorithms for aeroelastic simulations are currently under development as possible solutions to these requirements (Sabater et al. 2022; Zahn and Breitsamter 2022). Recently there has also been an increasing interest in utilising quantum computation and tensor network approaches (both on classical and quantum hardware) for Machine Learning (ML) (Stoudenmire and Schwab 2016; Reyes and Stoudenmire 2020; Dilip et al. 2022; Shen et al. 2024; Huggins et al. 2019). Therefore, we investigated the prospect of using hybrid quantum tensor network based algorithms for aeroelastic problems.

✉ M. Lautaro Hickmann
lautaro.hickmann@dlr.de

Pedro Alves
pedro.alves@dlr.de

David Quero
david.queromartin@dlr.de

Friedhelm Schwenker
friedhelm.schwenker@uni-ulm.de

Hans-Martin Rieser
hans-martin.rieser@dlr.de

[1] Institute for AI Safety and Security, German Aerospace Center (DLR), Ulm and St. Augustin, Germany

[2] Institute of Aeroelasticity, German Aerospace Center (DLR), Göttingen 37073, Germany

[3] Institute of Neural Information Processing, University of Ulm, Ulm 89081, Germany

Published online: 10 November 2025

A wide variety of QML approaches employing quantum circuits with tunable parameterised gates, so called Variational Quantum Circuits (VQCs), have recently been proposed (Schuld et al. 2021). Quantum tensor networks for ML can be realised by VQCs using a tensor network inspired internal gate structure. Tensor Networks (TN) were initially developed to reduce the computational cost of lowly entangled multi-particle quantum states. Nevertheless, they are able to efficiently approximate a wide variety of large tensorial objects using a regular, less complex structure. Thus, providing a convenient approach to QML (Rieser et al. 2023).

While QML is generating increasing interest in certain heuristic cases where advantages are suspected, these approaches often struggle to be described in the language of quantum circuits (Huang et al. 2021). For realistic problems, we see the necessity to integrate quantum-enhanced or based approaches into practical pipelines. Therefore, our focus lies in developing an end-to-end trainable hybrid Quantum Tensor Networks (QTN) approach.

This work seeks to investigate the potential benefits and limitations of hybrid QTN methods for realistic aeroelastic problems, with a focus on understanding their capabilities and constraints rather than directly searching for quantum advantages. By doing so, we aim to provide a comprehensive framework for the development of more effective hybrid QTN algorithms, ultimately contributing to the advancement of machine learning techniques for complex aeroelastic simulations and informing future research directions in this field.

Our approach emphasizes in particular efficient data encoding into quantum circuits for hybrid QML methods. Efficient encoding is a crucial step in QML. Recently, two promising approaches have emerged: pre-training quantum circuits to approximately encode data (Shen et al. 2024), and using TNs based encoding techniques to exactly encode data into quantum circuits (Ran 2020). However, these techniques have previously been used as preprocessing steps, where data is encoded once and then used as input for trainable quantum circuits. In contrast, our approach integrates the TN-based encoding into a fully end-to-end trainable hybrid algorithm. This approach entails a TN decomposition of the classical step to quantum gates, as explained in Subsection 2.2.

Our setup enhances current QML algorithms by combining three key components: a trainable TN-based dimensionality reduction, TN-based data encoding, and a trainable TN-inspired VQC (Shen et al. 2024), as explained in Subsection 2.3. This integrated approach enables end-to-end training using a single classical optimiser, allowing us to solve regression and classification tasks. An additional major technical contribution is the inclusion into a thorough state-of-the-art machine learning pipeline, including optimisation tools, mini-batch training and hyperparameter search with cross-validation.

2 Methods

2.1 Aeroelastic application

Classical tensor networks have various application scenarios within aeroelastics (Batselier et al. 2017). Relevant applications include data-driven aeroservoelasticity or the computation of dynamic loads resulting in an airframe of a manoeuvring aircraft (Jia et al. 2022). Within this field, nonlinear effects originating either from the structural or aerodynamic counterparts or a combination of them are of importance. The overall goal is to derive models from data which are able to predict aeroelastic characteristics (including damping) and thus, the stability behaviour of the system (Böswald et al. 2017).

One problem of particular interest in aeroelasticity is the determination of the flutter stability. To determine the stability, the feedback interaction between the structure and the aerodynamic forces has to be considered including inertial and elastic forces. We use a simplified aeroelastic configuration including a low-dimensional aerodynamic model for investigating the potential of QML for estimating the flutter stability of the system, based on Quero et al. (2019).

The selected case comprises a typical aeroelastic section of a wing with three degrees of freedom including heave h (positive downwards), pitch θ (positive nose up) around the elastic axis location and an aileron rotation β (positive with trailing edge down) (Tewari 2015). No assumption regarding the flow physics has been made and thus the methods are entirely data-driven, similar to Rauseo et al. (2021). The corresponding aeroelastic equations of motion are given by

$$\begin{bmatrix} 1 & x_\theta & x_\beta \\ x_\theta & r_\theta^2 & r_\beta^2 + x_\beta (c-a) \\ x_\beta & r_\beta^2 + x_\beta (c-a) & r_\beta^2 \end{bmatrix} \begin{bmatrix} \left(\frac{1}{L_{\text{ref}}}\right) \frac{d^2 h}{dt^2} \\ \frac{d^2 \theta}{dt^2} \\ \frac{d^2 \beta}{dt^2} \end{bmatrix} + \begin{bmatrix} \omega_h^2 & 0 & 0 \\ 0 & r_\theta^2 \omega_\theta^2 & 0 \\ 0 & 0 & r_\beta^2 \omega_\beta^2 \end{bmatrix} \begin{bmatrix} \frac{h}{L_{\text{ref}}} \\ \theta \\ \beta \end{bmatrix} = \frac{1}{\pi \mu} \left(\frac{U_\infty}{L_{\text{ref}}}\right)^2 \begin{bmatrix} -c_l(t) \\ 2c_m(t) \\ 2c_\beta(t) \end{bmatrix}, \tag{1}$$

where the structural damping has been neglected. The aerodynamic forces acting upon the structure are represented by the lift coefficient c_l, the pitching moment at the elastic axis c_m and the hinge moment at the aileron hinge axis c_β. A set of parameters has been chosen to be constant and their values are specified in Table 1, where the non-dimensional distances are obtained upon dividing by the reference length L_{ref}. Table 2 shows the variation of parameters carried out for the applications described next.

Equation 1 cannot be directly numerically integrated in time, as the aerodynamic coefficients are provided in the frequency-domain. When considering incompressible two-dimensional unsteady potential flow, they are provided in the frequency domain as irrational functions of the frequency. Thus, a specific procedure is applied in order to transform it into a state-space representation (Quero et al. 2019), which can finally be numerically integrated in time with a common ordinary-differential equation (ODE) solver:

$$\frac{d}{dt}\left(\begin{bmatrix} \mathrm{u}_h \\ \frac{d\mathrm{u}_h}{dt} \\ \mathrm{x}_a \end{bmatrix}\right) = \mathrm{A}_{ae}\begin{bmatrix} \mathrm{u}_h \\ \frac{d\mathrm{u}_h}{dt} \\ \mathrm{x}_a \end{bmatrix}, \qquad \mathrm{u}_h = \mathrm{C}_{ae}\begin{bmatrix} \mathrm{u}_h \\ \frac{d\mathrm{u}_h}{dt} \\ \mathrm{x}_a \end{bmatrix}, \tag{2}$$

where $\mathrm{u}_h = [h/L_{\mathrm{ref}}\ \theta\ \beta]^T$ and x_a contains the resulting aerodynamic states. For details on the matrices A_{ae} and C_{ae} the interested reader is referred to Quero et al. (2019). Once written in this form, the eigenvalues of the matrix A_{ae} determine the flutter stability of the aeroelastic system, which is then dependent on the value of the parameters given in Table 2, provided the parameters in Table 1 haven been fixed.

The goal of this application case is to apply hybrid quantum algorithms to the tasks of stability classification based on time series and regression of parameters from a time series. The stability of the system described in Eq. 1 is considered when subjected to non-zero initial conditions. In particular, the initial value of the first state component corresponding to a heave displacement is set to 1. Note that the physical magnitude is not of relevance here, as Eq. 2 is linear with respect to the states. Four representative time histories are provided in Fig. 1, where two stable and unstable cases each are represented for different combinations of the parameters (a, μ, U_∞), the three time series describe the response on each degrees of freedom (h, θ, β).

Table 1 Constant parameters

Description	Parameter	Value
Reference length	L_{ref}	0.5 (m)
Uncoupled heave natural frequency	ω_h	50 (rad/s)
Uncoupled pitch natural frequency	ω_θ	100 (rad/s)
Uncoupled aileron natural frequency	ω_β	300 (rad/s)
Non-dimensional distance from e.a. to the airfoil c.g.	x_θ	0.2
Non-dimensional distance aileron h.a. to aileron c.g.	x_β	0.0125
Non-dimensional airfoil radius of gyration about e.a.	r_θ	$\sqrt{0.25}$
Non-dimensional aileron radius of gyration about aileron h.a.	r_β	$\sqrt{0.00625}$
Non-dimensional distance between the midchord and the aileron h.a.	c	0.5

Centre of gravity is denoted by c.g., elastic axis by e.a., and hinge axis by h.a

Table 2 Varied parameters

Parameter	Interval	Increment
Non-dimensional distance between midchord and e.a.	$a \in [-0.4, 0.4]$	$\bigtriangleup a = 0.1$
Mass ratio	$\mu \in [10, 50]$	$\bigtriangleup \mu = 0.1$
Airspeed	$U_\infty \in [150, 350]$ (m/s)	$\bigtriangleup U_\infty = 1$(m/s)

Elastic axis is denoted by e.a

2.2 TN based preprocessing and encoding

Classical Tensor Networks (CTNs) have various applications within aeroelastics, such as aeroelastic system identification. The goal is to derive data driven models which enable the prediction of aeroelastic characteristics including the stability behaviour of the system (Böswald et al. 2017).

Tensor Networks were originally conceptualised to facilitate the simulation of Quantum Many-Body Systems by reducing the amount of correlations inside a quantum state (Wall et al. 2022). They are a set of tensor objects connected with each other in a specific layout through index contraction. Several different regular tensor network layouts with varying dimensionality have been studied. The most common ansatz is the Matrix Product State (MPS) (Fannes et al. 1992) shown in the first row of Fig. 2. Other layouts are tree tensor networks (TTN) (Murg et al. 2015), MERA networks (Vidal 2007), which are trees with entangled branches and two-dimensional PEPS networks (Sierra and Martin-Delgado 1998).

The aforementioned TNs, are a powerful tool for representing complex data structures, enabling the efficient manipulation of classical and quantum systems. By applying Tensor Network Operators (TNOs), it is possible to perform operations on data in TN format, effectively modifying the underlying structure. TNOs represent local linear transformations, akin to matrix multiplications, which can be applied to specific sections of the TN. For instance, MPSs can be transformed using Matrix Product Operators (MPOs), which are defined by introducing an additional free index at each site, where one is considered the upper index (free input index) and lower index (free output index). By contracting the input indices of the MPO with those of the MPS, a new, transformed MPS is produced from the free output indices, allowing for efficient manipulation and analysis of complex data structures.

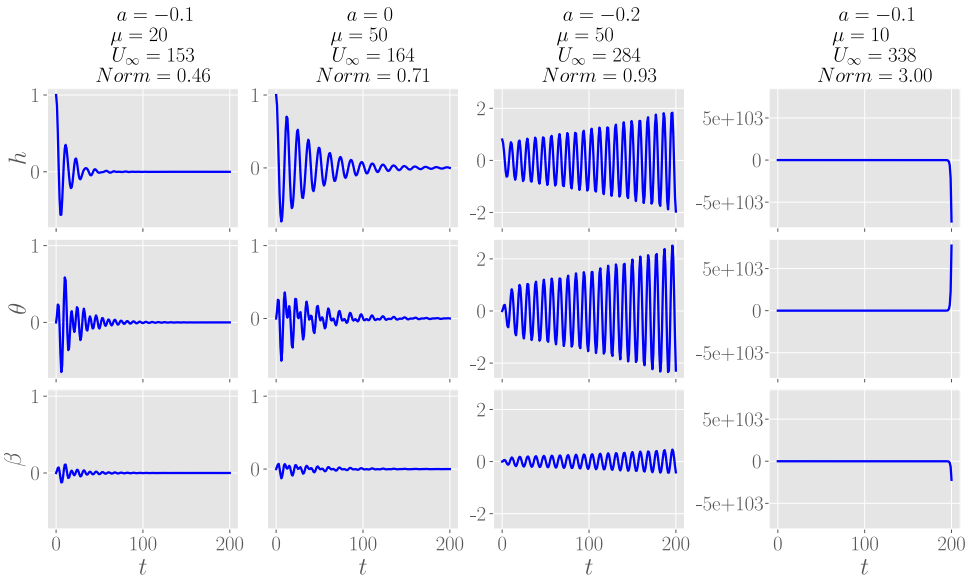

Fig. 1 Prototypical aeroelastic time series responses for each degree of freedom for four sets of aeroelastic parameters. The two on the left are stable and the two on the right unstable response

When choosing a tensor network type for a specific task, it is crucial to take the scaling behaviour of the task into account. While one-dimensional data like time series are handled well using MPS, images require a two-dimensional scaling of the information entropy in the worst case. In this study, we focus on time series therefore a one-dimensional MPS layout is well suited. In an MPS two additional hyper-parameters can be chosen: the bond dimension which is determined by the number of qubits passed on from each node to the next, and the number of data qubits that are passed to the circuit at each iteration.

A Quantum Tensor Networks (QTN) is a TN that represents a compressed version of a quantum state implying that the TN is primarily under the normalisation condition. One additional property relevant for this work is that a canonically gauged one-dimensional QTN can be mapped to a quantum circuit (Ran 2020; Liu et al. 2019; Huggins et al. 2019). Once in canonical gauge, the majority of its tensors are isometries which can be converted into unitary gates with some linear algebra kernel acquisition technique (Ran 2020). In other words, the output of such circuit of unitaries reflecting the one-dimensional layout, is the quantum state encoded by the QTN. Figure 2, shows the mapping of a canonised MPS, to a quantum circuit.

TNs already found their way into applications within classical machine learning (Reyes and Stoudenmire 2020). For instance, here a CTN can represent an input vector, a linear operator or encode non-linear functions while benefiting from local operators that preserve a compressed representation of the problem at hand. At the expense of a normalisation constant, a CTN can be transformed into a QTN, for it to subsequently be mapped to a quantum circuit.

For our use case, the input data consists of a three-dimensional time series described in Subsection 2.1. The time series have a wide range of values with the converging ones usually being in the range of ± 1 but the diverging one can take values over $\pm 5 \times 10^{100}$. As a first step we normalise

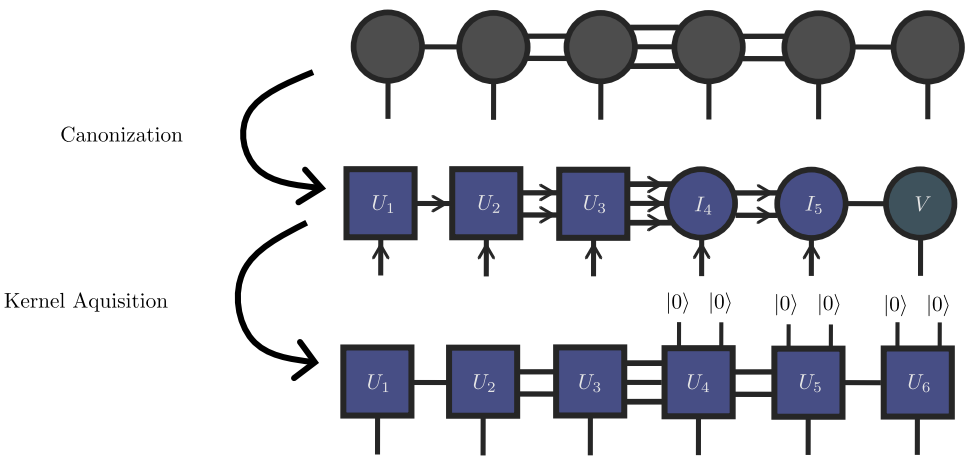

Fig. 2 QTN quantum circuit mapping example of a MPS representing a quantum state where the bond dimension increases exponentially to the centre bond. Here, the main phases are highlighted: the canonisation already leaves 3 tensors as unitary gates (blue squares), 2 tensors as isometries (blue circles) and 1 orthogonality centre representing essentially a normalised vector (green circle); the kernel acquisition step ensures that the full unitary gates can be found and the beginning of the quantum wires can be assigned with the zero-state $|0\rangle$

each group of time series as one, i.e. concatenate all three time series, normalise the resulting one and then separate them again. This way the amplitude relation between each dimension of the time series remains unchanged. Each data point, i.e. each group of three time series for a set of aeroelastic parameters (a, μ, U_∞), is normalised independently and the norms (re-scaled to $[0, \pi]$) are saved to be utilised as an input to the quantum circuit.

As quantum mechanics is a linear theory, nonlinear behaviour can only be introduced by interactions with the classical environment, e.g. by carrying out measurements or during encoding of the normalised inputs x using some encoding map $\Phi(x)$ (Yan et al. 2015). Finding the right encoding strategy for a QML application is often a difficult task. A variety of methods have been developed which lie between the two extremes: "qubit efficient", realised by amplitude encoding, and "gate efficient", realised by binary encoding. In this work, we decided to use a tensor network operator to reduce the dimensionality of the data and learn the encoding to be deployed on the quantum circuit, as shown in Fig. 3. Additionally, tensor networks mapped to quantum circuits can have qubit efficient representations (Rieser et al. 2023).

Due to the one-dimensional structure of the time series, tensor networks and specifically MPS are well suited to express this type of data (Rieser et al. 2023). To improve the compatibility of our data with TNs, we preprocessed the original time series by upsampling it from 201 time steps to $3^5 = 243$ using smoothing B-splines. This upsampling enables an efficient decomposition into a 5-node MPS with free indices of dimension 3, allowing for a compact representation of the data. An additional free index is introduced to select one of the three time series, effectively encoding the time series dimension. The preprocessing steps are illustrated in the first row of Fig. 3.

This MPS is contracted with a subsequent trainable MPO that aims to reduce the dimensionality of the input MPS and

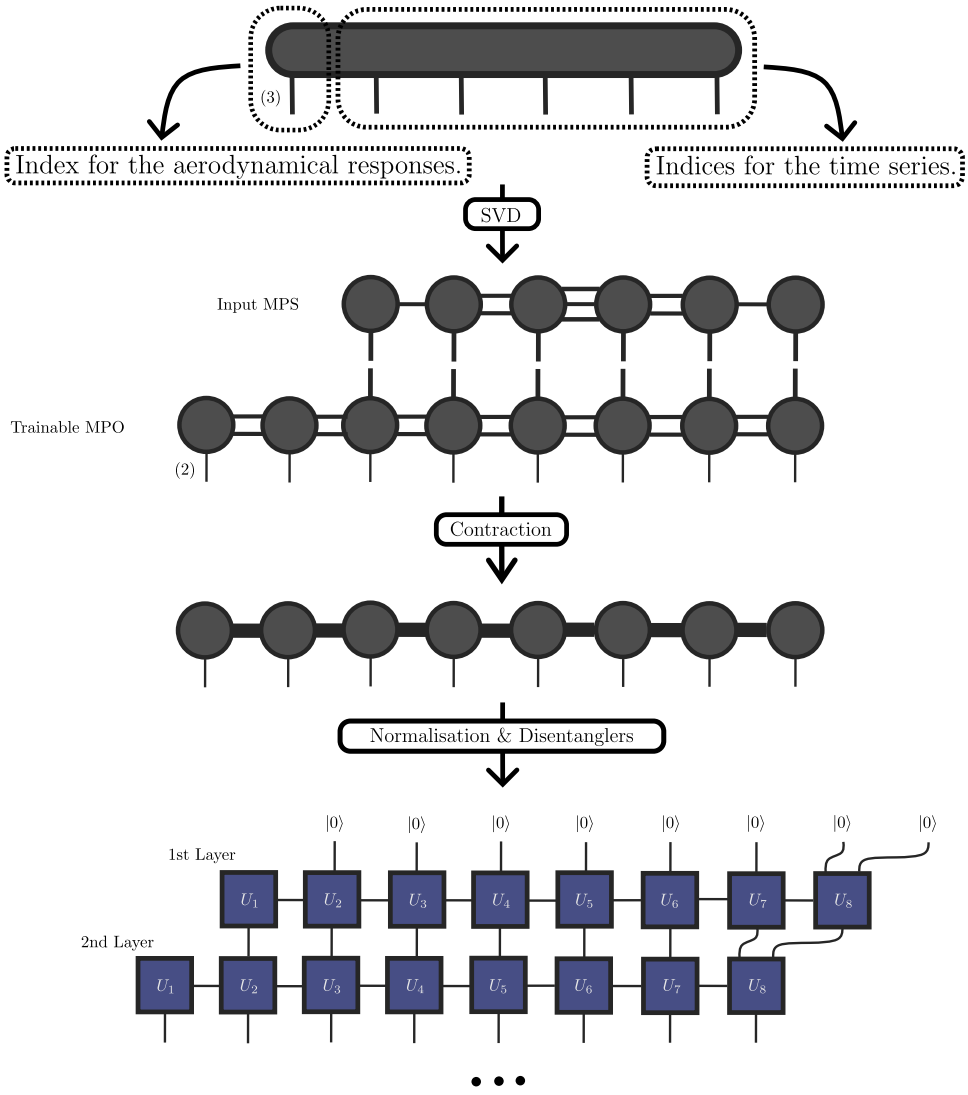

Fig. 3 Preprocessing and encoding structure. First we encode the three time series comprising each data point into an MPS. We then apply an MPO based dimensionality reduction and non-linearities. Finally we disentangle the resulting TN into a quantum circuit

learns the optimal encoding scheme to the quantum circuit. For this purpose, the MPO has 6 free input indices of dimension 3 and 8 free output indices of dimension 2. The dimension of the MPO's internal bonds can be adjusted to carry more trainable parameters and potentially add more correlations to the data, thus being considered a hyper-parameter. Naturally, the output of this contraction is another MPS with 8 free indices of dimension 2. To improve the expressivity, we applied a *tanh* nonlinearity on the resulting MPS parameters before normalisation.

Since one of the objectives of this preprocessing step is to learn what to encode in the quantum circuit, this output CTN still needs to be converted into a QTN which can be done by normalising it (Dborin et al. 2022). We then used the technique shown in Ran (2020), to map it into a quantum circuit. This technique adds layers of cascading 2-qubit gates that have the ability to progressively disentangle the state represented by the output of the MPO (except for a normalisation constant) to the zero-state $|00...\rangle$, hence being called Matrix Product Disentangler (MPD). From another perspective, inverting the order of these layers and transpose-conjugating every unitary, the layers will progressively entangle the state $|00...\rangle$ until the desired state is reached. The more layers there are, the more correlations/entanglement can be achieved in the state. Therefore, the number of MPD layers is considered a hyper-parameter in this set-up for gradually adjusting the entanglement of the input state to the VQC circuit. The complete procedure is shown in Fig. 3.

2.3 TN inspired VQC classifier

Once we have the data encoded into a quantum circuit, the next step is to process it using QML. Quantum computation in general and QML in particular are still in very early stages of development. Today, most quantum algorithms are written on a circuit level. When designing a quantum circuit, choices on a very basic level must be made, e.g. the data encoding circuit, entangling schemes and the measurement processes. As it is not clear to date which choices are most relevant for the quantum machine learning application we carried out a comprehensive hyperparameter search.

A VQC is a quantum circuit with gates that feature tunable parameters, usually rotational gates. A general variational unitary can be decomposed into a combination of rotation and entangling gates like the CNOT. A common category of VQC architectures for machine learning are layered VQCs. Here, the circuit consists of encoding blocks that map the data to the circuit, and variational blocks which entangle the qubits and introduce the optimisable parameters. To increase expressivity, these blocks are executed repeatedly (Schuld et al. 2021).

Another approach is to employ a QTN for machine learning using parameterised gates, often also called TN inspired ansatz. It is a variety of VQC that carries a tensor network based internal gate structure (Dilip et al. 2022). The construction of the tensor network approach using "states" and "operators" makes it straightforward to translate the concept to quantum computation as seen in Fig. 4. Both are realised by a set of parameterised multi-qubit gates where the only difference between states and operators is whether the gates have only incoming or outgoing free bonds or both.

To make use of the capabilities of quantum computers, the individual qubits have to be entangled by using multi-qubit gates. These gates have several free parameters that can be used to define how the incoming data is processed. When kept trainable, they can act as parameters of machine learning algorithms.

As shown in Fig. 4, we first encoded the data by using one or more layers of arbitrary unitaries derived from the preprocessing MPS through the disentangling process. Additionally, we encoded the norm of the original time series obtained in the first normalisation step and the norm of the

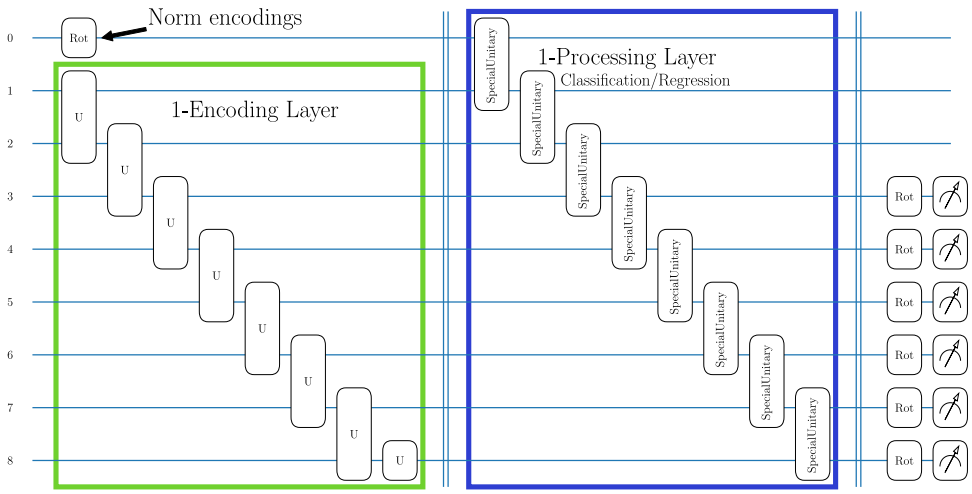

Fig. 4 Typical layout for an MPS inspired VQC for the regression task measuring multiple qubits. Here including Norm encoding and using one encoding and one classification layer. Lastly a trainable measurement layer is applied. Similar to Fischbach et al. (2025)

resulting MPS as parameters of a general rotation on the first qubit. Since otherwise, the information on the amplitude relationship between time series would have been lost.

After the encoding, we applied a MPS inspired structure constructed from iterated layers of two qubit gates (Dilip et al. 2022; Jobst et al. 2024; Shen et al. 2024). For the two qubit gates, we considered strongly entangling layers on two qubits (Schuld et al. 2020) with 6 trainable parameters per gate and general $\mathrm{SU}(4)$ unitaries (Wiersema et al. 2024) with 15 trainable parameters per gate. For QTN, some authors recommend shuffling the remaining virtual qubits before measurement using a single layer of general rotations on the qubits being measured (Shen et al. 2024).

The measurement which later can be interpreted as the result of our machine learning algorithm can be performed in several ways. For classification problems, the simplest way is to chose one qubit as the output qubit. We measured the probabilities of the basis states on the last qubit and interpreted them as class probabilities, which were then compared against one hot encoded binary stability labels.

For the regression task, we measured either the expectation values of individual qubits or tensor product of pairs of qubits to predict the aeroelastic parameters. To make the comparison possible, we re-scaled the aeroelastic parameters to the ± 1 range independently per feature. We investigate univariate regression by predicting each of the aeroelastic parameters with different models and multivariate regression by predicting all aeroelastic parameters at once. For the latter case, we measured either three Z_i observables or three tensor products $\{Z_i \otimes Z_j\}_{i \neq j}$ observables on non-overlapping pairs of qubits and interpreted them as elements of the target vector. Similarly for the univariate setup, we only measured one observable.

Besides the architecture of the quantum circuit, other parameters can be adjusted. There are several different classical optimisers that can be uses to train hybrid-quantum circuits. The parameters of the MPO and VQC where then jointly optimised using a classical optimiser through back-propagation with auto-differentiation. We used the well known Adam optimiser (Kingma and Ba 2017), which uses global gradients. As loss functions, we used cross entropy for the classification tasks, and the Huber loss (Huber 1964) for the regression tasks.

The computations were performed using the Quimb (Gray 2018) library for TNs, PennyLane (Bergholm et al. 2022) for quantum circuits and simulations, and JaX (Bradbury et al. 2025) for the ML components. For statistical robustness we used cross validation, using the ShuffleSplit method. We used the implementations provided in the scikit-learn library (Pedregosa et al. 2011). Lastly we used the Optuna (Akiba et al. 2019) hyperparameter search framework.

For each hyperparameter configuration, a 5-fold cross validation was carried out, using the methods previously explained. At the end of each training, the maximum scores per fold over all previous epochs of the metrics were averaged and used as the objective value for the hyperparameter search algorithm. We used the well known F_1 score for the classification task and R^2 score for the regression task.

After conducting the hyperparameter search, we retrained the best configuration for each task using the complete training set as folds, i.e. we ran the training 5 times with different random seeds on the complete training dataset (training + evaluation datasets used for the hyperparameter search), and tested the trained models on the test set.

3 Results and discussion

After conducting an exhaustive hyperparameter search and retraining the top-performing quantum models, we analysed the results and evaluated various metrics to identify potential bottlenecks and areas of improvement in order to gain insights into the performance of our hybrid QTN approach. Furthermore, we also compared the results to those obtained using Multilayer Perceptrons (MLPs) with a comparable number of trainable parameters using two-layers. We begin by presenting the results achieved by the quantum networks, followed by a comparison to approximately equivalent classical counterparts.

We found that our hybrid TN inspired algorithm could easily solve the binary time-series classification, achieving a maximum F_1-score of well above 0.9, averaged over 5 repeated training runs. The best model achieved a F_1-score of 0.998, as shown in the Confusion Matrices (CMs) in Fig. 5. The models generalised very well, and we observed no overfitting. We carried out a very limited hyperparameter search, since we quickly found well performing configurations. As it can be observed in the CMs, all training seeds converged towards good results.

While doing the hyperparameter search, we could observe that many configurations were unstable, achieving significantly different results for each fold. Overall, several configurations achieved good results, the best model used only a small MPO bond dimension of 2 and only one disentangling layer, but needed four TN inspired quantum classification layers. This hints at the majority of the processing being done on the VQC side, for a highly compressed and potential low-entanglement representation obtained through the utilised MPO.

Our analysis of the training behaviour revealed that most runs converged rapidly, with most models achieving optimal performance within 5 epochs, with the amount of gradients updates depending on the utilised batch size. Although this might suggest the presence of barren plateaus, a closer

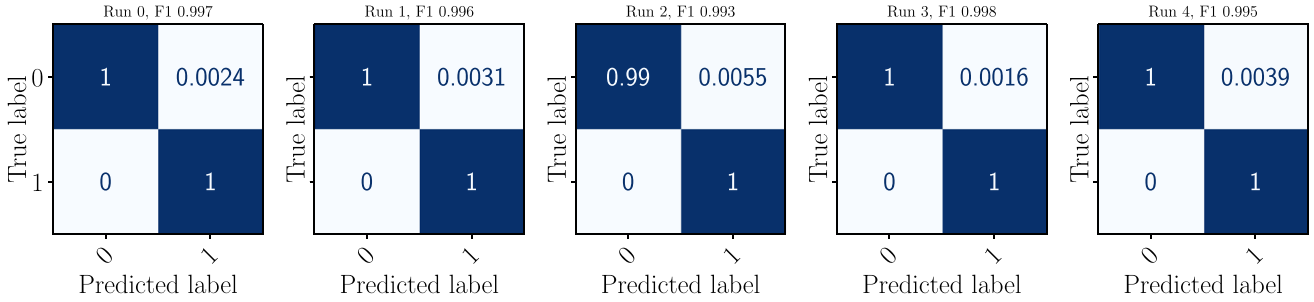

Fig. 5 F1 confusion matrices on the test set for the best model for the classification task, for 5 different random training seeds

examination of the gradient variance showed that the quantum part of the model exhibited a tendency towards zero, but did not completely disappear. In contrast, the gradients of the MPO displayed more instability, with some values converging towards zero before spiking up and then returning to near zero. Despite this unusual behaviour, the classifier achieved outstanding results. This behaviour calls for an in depth analysis of the loss landscape, and in particular the relations between the hyperparameters and the trainability of the models.

The test behaviour over training epochs for the best hyperparameter configuration on the classification task, as shown on the left side of Fig. 6, demonstrate rapid convergence, with approximately 2 epochs required to achieve stable performance. This corresponds to around 114 mini-batch gradient updates, for the selected batch size of 128. Notably, the models' performance did not degrade over prolonged training. Furthermore, the initial spread of F1 values at epoch 0, reflects the impact of different weight initialisation. However, as training progressed, all models converged to remarkable results, suggesting that the optimisation process was robust and effective.

For the regression task, we used the same TN inspired VQC algorithm as for classification, with adjustments for the number of measured qubits and the type of measurement carried out. As described in Subsection 2.3, the main difference is that we measured expectation values instead of basis state probabilities. Since we require more granular control over the predictions.

Overall, the regression task exhibited less stability, particularly for multivariate regression, as shown in the right side of Fig. 6. The univariate regression results were significantly better, as depicted in Figs. 7 and 8.

A notable observation from the hyperparameter search was that some multivariate configurations achieved near-optimal performance on specific target parameters (one of a, μ or U_∞), comparable to those of univariate models. However, these models often predicted a single target dimension well while failing to accurately predict others. Nevertheless, the best multivariate model achieved a more balanced performance, albeit with a slightly lower R^2 score per target dimension. This suggests that the model was able to capture only certain aspects of the data, but not everything necessary for performing all three regressions at once.

In contrast, the models trained in a univariate fashion, i.e. to predict only a single target dimension, achieved outstanding results. Notably, different hyperparameter configurations yielded the best results for different target dimensions, suggesting that the chosen ansatz lacked flexibility. This may be attributed to the lack of trainable non-linearities.

As observed in Figs. 7 and 8, the repeated runs for different seeds exhibited good convergence, but with a considerable spread. This variability implies that better initialisation techniques are needed. Our results indicate that using a noisy identity initialisation led to more stable results compared to a random uniformly distributed initialisation. However, a more

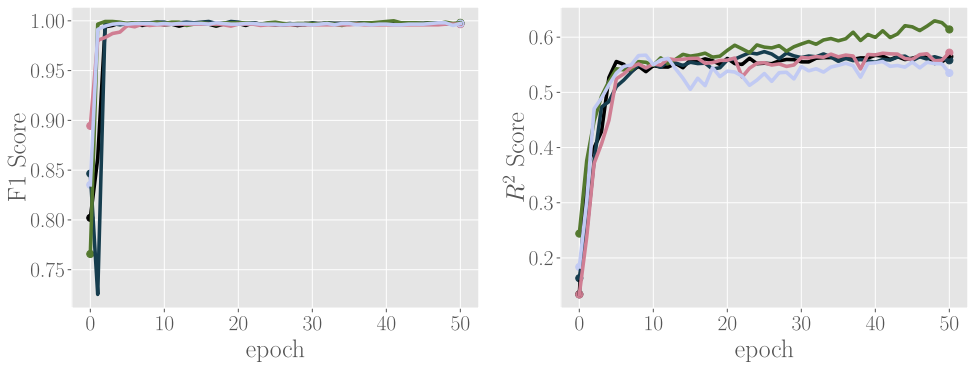

Fig. 6 Test F1 score (left) and test R^2 score (right) values over training epochs for 5 different training seed of the best classification (left) and multivariate regression (right) model

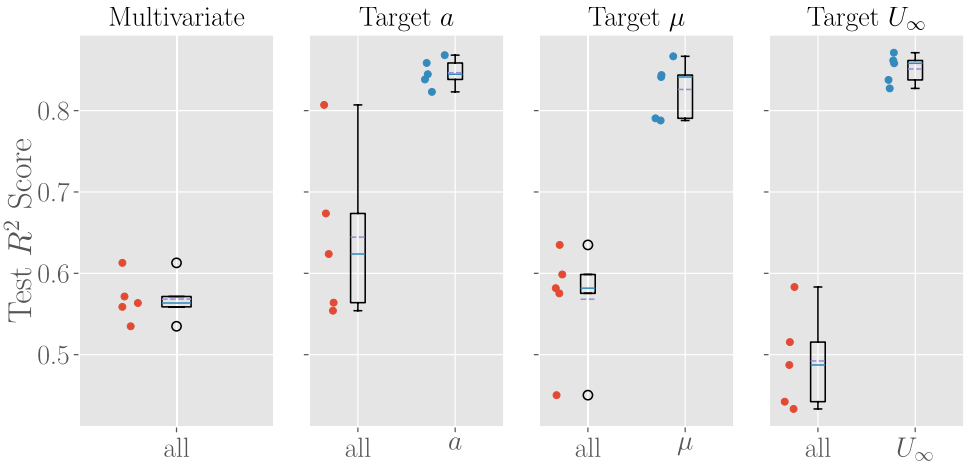

Fig. 7 Test regression score for the best model for the multivariate and univariate regression task, averaged over 5 runs of each model. Each column represents different prediction targets, and the labels at the bottom represent the targets the model was trained to predict

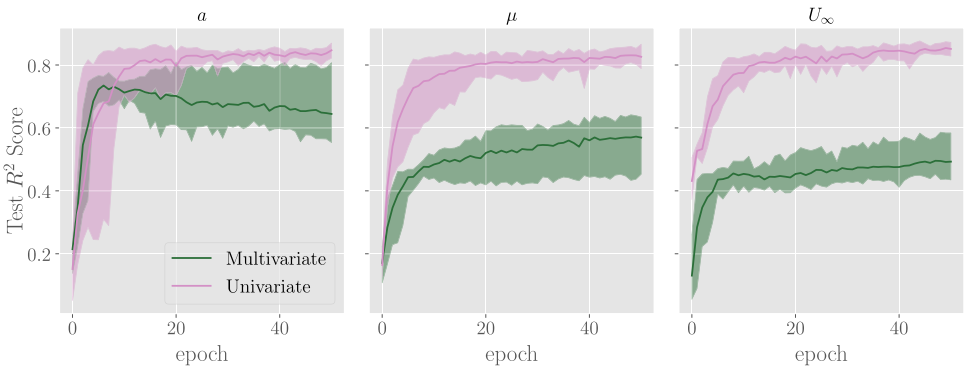

Fig. 8 Test regression score for the best model for the multivariate and univariate regression task over training epochs, averaged over 5 runs of each model. Each column represents different prediction targets

in-depth analysis is required to isolate the factors contributing to this behaviour.

An interesting observation was made when examining the test R^2 scores during training of the best selected model, shown in Fig. 8. The multivariate model initially performed well on the a-dimension, but then its performance degraded. Conversely, the other two dimensions (μ and U_∞) showed improvement, suggesting that the model was learning to balance its predictions. This phenomenon may indicate that the model was initially over-specialised to one dimension and then generalised to the others. This behaviour could indicate at a lack of expressibility of the model, or the hardness of the multivariate regression task.

The multivariate models exhibited a significantly wider spread across the 5 different training seeds compared to the univariate setup. Notably, the two setups displayed opposing trends in variance over time. The multivariate model showed relative stability at epoch 10, followed by a significant divergence, particularly for the a and μ dimensions. In contrast, the univariate models demonstrated a reduction in variance over advanced training epochs.

The difference on the R^2 scores for multivariate regression and the reason the hyperparameter configurations differ for each univariate setup can be explained by the different granularity of each component: a has 9 distinct values, μ has 5 and U_∞ has 201. The score difference between a and μ could be explained by the former having enough values to better cover the normed target value range, resulting in smaller errors between predictions and targets.

The regression task proved to be considerably more challenging than the classification task. The prediction of discrete variables with multiple possible values using expectation values of observables is a more complex task than binary classification. The model struggled to accurately predict the values, and the random initialisation of the models resulted in significant variability in performance.

In particular, we observed that for the cross validation some runs of the same configuration worked out well while others did not converge to a good score. This behaviour could be explained by instabilities arising from the data points randomly selected for each cross-validation fold, i.e. the failure to generalise from specific subsets of the dataset. This may be mitigated by using more than five runs, i.e. using more folds, per configuration. Although, steeply increasing the resources requirements accordingly. The second possibility would be to theoretically investigate the loss landscapes and initialisation of such quantum circuits and use these results to begin with better initial conditions.

The analysis of hyperparameter importance revealed similarities between the classification and regression tasks, as well as between the two regression setups. Overall, the interpretation of the hyperparameter search results was challenging due to the high spread for single hyperparameter values. This lead to the conclusion that not single, but combinations of hyperparameters have a significant impact on the model's performance.

We used four parameter importance estimators, namely fANOVA (Hutter et al. 2014), Shapley TreeExplainer (Lundberg et al. 2020), Mean Decrease Impurity (MDI) (Agarwal et al. 2023), and PED-ANOVA (Watanabe et al. 2023) to select a subset of most important hyperparameters. The results showed that in almost all cases the number of quantum processing layers, the learning rate, and the chosen quantum ansatz were the most significant contributors to the model's performance, except for the univariate a model were instead of quantum layers the MPO bond dimension played a role.

These observations suggest that the VQCs play a crucial role in determining the model's performance. Two possible interpretations are that either the VQCs are doing most of the processing, and their performance is heavily influenced by these hyperparameters. Or, alternatively, the VQCs may be a bottleneck towards the model's information processing capacity. Especially for the more complex regression tasks since the classification tasks could be perfectly solved.

On the other hand, the results also suggest that the MPO parameters have a limited impact on the model's performance. This could either mean that all tested configurations discard too much data leading to the quantum part having to do the heavy lifting, or that they provide enough flexibility to adapt to the needed information content.

The analysis highlights the importance of hyperparameter tuning and the need for further investigation: An in-depth ablation study would be necessary to fully understand the impact of these hyperparameters on the model's performance. Finding out what would be the best way to improve the synergies of the classical, TNs, hyperparameters with the quantum circuit hyperparameters remains challenging, and an in depth information theoretical analysis is needed.

To compare the performance of our quantum approach with a classical setup, we carried out a comprehensive evaluation after selecting and retraining the best quantum models. Ensuring a fair comparison between quantum and classical models is a challenging task, which remains an open question in the field. To make the comparison as fair as possible, we used the same training setup up to the disentangling step of the framework described in Fig. 3. We chose a two-layer Multilayer Perceptron (MLP) architecture for the classical model allowing for non-linear hidden activations. We adjusted the dimension of the hidden representation to achieve a comparable number of trainable parameters with the selected quantum model for each task. The output activation functions used were *softmax* for classification and *tanh* for regression, analogous to the probabilities and expectation values obtained from the quantum model.

For the classical network, we used the same MPO setup but contracted the final MPS instead of building disentangler layers. This yielded a 256-dimensional feature vector. Similar to the quantum case, we added the norm of the original time series and the norm of the MPO, resulting in a 258-dimensional input vector for the classical network. By doing so, we ensured that the number of trainable parameters for the MPO remained equal for both the quantum and classical networks. Furthermore, the output dimension of the last layer was adapted to accommodate the specific task requirements, with a single output for classification and univariate regression, and a three-dimensional output for multivariate regression, comparable to the quantum model as described in Subsection 2.3.

After conducting a hyperparameter search, we selected the best models with similar numbers of trainable parameters. Notably, all tasks exhibited optimal performance when using an *elu* activation function (Clevert et al. 2016) in the hidden layer. For the binary classification task, the quantum models achieved perfect results, and consequently, no significant differences were observed when comparing with the classical networks. In contrast, the regression tasks exhibited mixed behaviour, as illustrated in Fig. 9.

A key difference between the classical and quantum models is the reduced spread of results obtained from the classical networks, which highlights the need for improved initialisation and convergence analysis of quantum models. The classical networks demonstrated slightly better performance overall. Interestingly one can observe a big improvement in the a and μ dimensions, but worse results on the U_∞ dimension when analysing each dimension of the multivariate regression individually. On the other hand, the multivariate regression results were better for the classical model. The general behaviour and "hardness" of each dimension were similar for both paradigms. For univariate regression, the classical methods showed slightly better results across all dimensions.

It is worth mentioning that for the univariate regression of the a target, we could not obtain a classical model with a comparable number of trainable parameters due to the extremely small size of the quantum model. As a result, the improved performance of the classical setup required approximately $\approx 70\%$ more trainable parameters. Nevertheless, in the hyperparameter search for the quantum model, there were setups using more trainable parameters that did not achieve better results.

A fundamental challenge in comparing the two paradigms lies in how information is transferred from the TN based preprocessing to the respective neural networks. Since we contracted the resulting MPS for the classical network, this can be viewed as equivalent to having an exponentially large

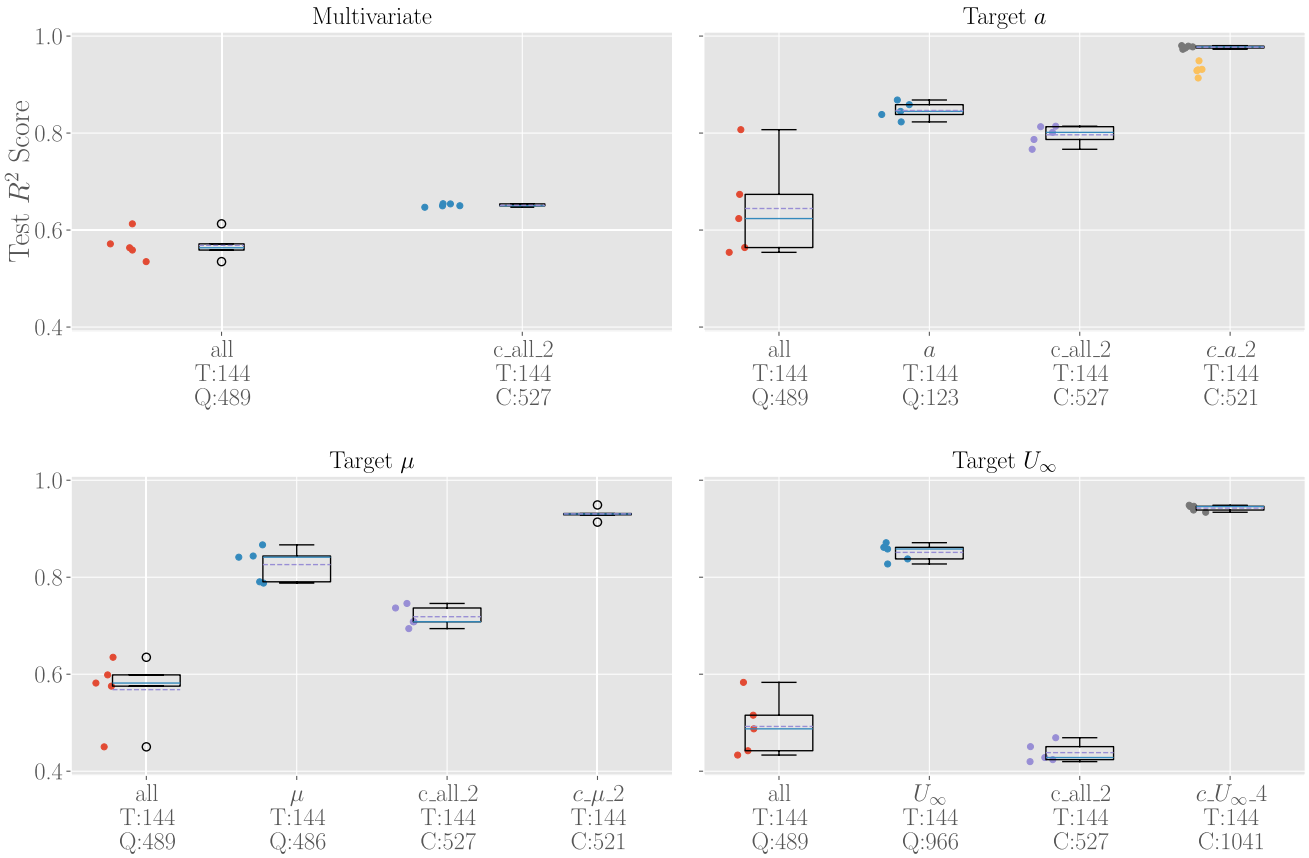

Fig. 9 Test regression scores for the best quantum and comparable classical models on the multivariate and univariate regression tasks, averaged over 5 runs per model. The classical models (marked with a $c\_$ preposition) were selected to have a similar number of trainable parameters to the best quantum models to enable a fair comparison. For the classical models, the $\_i$ post-fix denotes the input dimension of the second layer. Each subgraph shows one of the regressed aeroelastic parameters. The numbers below the graphs show the number of trainable parameters: T refers to the trainable MPO, Q denotes the VQC for the quantum networks, and C for their classical counterparts used

number of disentangling layers, whereas our quantum setup used only 2 to 8 layers due to computational constraints. The implementation of additional disentangling layers for the quantum setup was prohibitively expensive, representing a significant bottleneck in our current approach. Addressing this limitation will be a key focus of future work.

Overall, we show that realistic regression problems can be tackled by hybrid QML approaches and the results provide valuable insights into the importance of hyperparameter tuning and the challenges of optimising VQCs. Further research is needed to fully understand the impact of these hyperparameters on the model's performance.

4 Conclusion and future work

In conclusion, our study demonstrates the successful application of hybrid quantum tensor network-based algorithms to aeroelastic problems. By integrating three key components - trainable TN-based dimensionality reduction, TN-based data encoding, and a trainable TN-inspired VQC - we enable end-to-end training using a single classical optimiser, eliminating the need for pre-training circuit representations for each data point.

We could solve the binary classification task perfectly, and achieved promising results for the time series multivariate and univariate regression tasks. Although, the optimal choice of hyperparameters remains a challenge.

The main limitation of our current setup lies in the computationally prohibitively expensive implementation of disentangling layers. The resources needed for each subsequent layer increase very rapidly, and since we are creating them in each forward pass and using cross-validation in parallel, this severely limits the number of disentangling layers that could be used. This leads to the VQCs only using an estimation of the MPO, which could represent a bottleneck in the expressivity of our framework. However, since it is trained end-to-end, it is possible that the system learns to pass the

necessary information despite this limitation. To address this issue, we propose exploring more efficient methods for implementing disentangling layers or developing novel algorithms for encoding TNs into VQCs. Additionally, we plan to conduct an in-depth analysis from an information theory point of view to better understand the effects of the encoding setup.

Further future research directions include conducting an in-depth ablation study to explore the relationship between encoding parameters, such as bond dimension and number of disentangling layers, and the expressivity of the circuit. The classification task results suggest that entanglement/correlations in the compressed encoding data may not be essential for the classifier. However, it is unclear what portion of the Hilbert space the classifier/regressor accesses, potentially leaving room for improvement with encodings that offer more entanglement - a unique property of quantum computing. To clarify this point, future work will focus on analysing the TN representation of the processing layers, including examining the bond dimension of the TN representation of the quantum circuit, which provides insights into the amount of correlations present in the system.

To further improve the performance of our hybrid model, we recognise the need to introduce more non-linearities into the hybrid model. This is a natural problem that occurs in quantum mechanics, being, at its core, a linear theory. To address this, we propose exploring techniques that interact with a classical environment, such as encoding, data re-uploading (Pérez-Salinas et al. 2020), natural noise models, or analogue mode operations present in Noisy Intermediate Scale Quantum (NISQ) devices. These approaches have the potential to add sufficient non-linearities to the hybrid QML pipeline, leading to improved performance on aeroelastic problems.

In this work, we aimed to investigate the potential benefits and limitations of hybrid QTN methods for realistic aeroelastic problems, with a focus on understanding their capabilities and constraints rather than searching for quantum advantages. While our results show similar behaviour between quantum and classical methods, we did not find potential quantum advantages in this study. However, these advantages prove to be notoriously hard to find in realistic applications. Therefore, our main goal lies in presenting a novel framework for end-to-end hybrid QTNs, which can be applied to a wide range of tasks and use cases. To fully realise the potential of these methods, we plan to do an in depth analysis of the scalability and performance of our approach in large realistic datasets. This may ultimately lead to identifying quantum advantages, particularly in terms of the amount of training data required and the representation power of Quantum Neural Networks.

Overall, this study provides a foundation for future research in hybrid quantum tensor network-based algorithms for aeroelastic problems. By addressing the challenges and limitations identified in this work, we can unlock the full potential of these algorithms and explore their applications in more complex and realistic scenarios.

Acknowledgements We would like to express our sincere gratitude to Steffen Leger, Gustav Jäger, Andrew Barlow, and Krzysztof Bieniasz for their valuable contributions.

Author Contributions Conceptualization: M.L.H., D.Q., H-M.R; Methodology: M.L.H., P.A., H-M.R; Formal analysis and investigation: M.L.H., D.Q., H-M.R; Writing - original draft preparation: M.L.H.; Writing - review and editing: M.L.H., P.A., D.Q., F.S., H-M.R; Funding acquisition: D.Q., H-M.R; Data curation: D.Q.; Supervision: F.S., H-M.R.

Funding Open Access funding enabled and organized by Projekt DEAL. M. Lautaro Hickmann received funding from the DLR Quantum Fellowship Program.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing Interests The authors declare no competing interests.

References

Agarwal A, Kenney AM, Tan YS, Tang TM, Yu B (2023) MDI+: A Flexible Random Forest-Based Feature Importance Framework

Akiba T, Sano S, Yanase T, Ohta T, Koyama M (2019) Optuna: A Next-generation Hyperparameter Optimization Framework. https://doi.org/10.48550/ARXIV.1907.10902

Batselier K, Chen Z, Wong N (2017) Tensor network alternating linear scheme for MIMO volterra system identification. Automatica 84:26–35. https://doi.org/10.1016/j.automatica.2017.06.033

Bergholm V, Izaac J, Schuld M, Gogolin C, Ahmed S, Ajith V, Alam MS, Alonso-Linaje G, AkashNarayanan B, Asadi A, Arrazola JM, Azad U, Banning S, Blank C, Bromley TR, Cordier BA, Ceroni J, Delgado A, Matteo OD, Dusko A, Garg T, Guala D, Hayes A, Hill R, Ijaz A, Isacsson T, Ittah D, Jahangiri S, Jain P, Jiang E, Khandelwal A, Kottmann K, Lang RA, Lee C, Loke T, Lowe A, McKiernan K, Meyer JJ, Montañez-Barrera JA, Moyard R, Niu Z, O'Riordan LJ, Oud S, Panigrahi A, Park C-Y, Polatajko D, Quesada N, Roberts C, Sá N, Schoch I, Shi B, Shu S, Sim S, Singh A, Strandberg I, Soni J, Száva A, Thabet S, Vargas-Hernández

RA, Vincent T, Vitucci N, Weber M, Wierichs D, Wiersema R, Willmann M, Wong V, Zhang S, Killoran N (2022) PennyLane: Automatic differentiation of hybrid quantum-classical computations . arXiv:1811.04968

Böswald M, Schwochow J, Jelicic G, Govers Y (2017) Recent developments in operational modal analysis for ground and flight vibration testing. In: Proceedings of the International Forum on Aeroelasticity and Structural Dynamics (IFASD 2017). https://elib.dlr.de/114142/

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., Zhang, Q.: JAX: composable transformations of Python+NumPy programs (2025). http://github.com/jax-ml/jax

Clevert D-A, Unterthiner T, Hochreiter S (2016) Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs) . arXiv:1511.07289

Dborin J, Barratt F, Wimalaweera V, Wright L, Green AG (2022) Matrix product state pre-training for quantum machine learning. Quant Sci Technol 7(3):035014. https://doi.org/10.1088/2058-9565/ac7073

Dilip R, Liu Y-J, Smith A, Pollmann F (2022) Data compression for quantum machine learning. arXiv:2204.11170v2

Fannes M, Nachtergaele B, Werner RF (1992) Finally correlated states on quantum spin chains. Commun Math Phys 144:443–490

Fischbach F, Rieser H-M, Sefrin O (2025) Encoding hyperspectral data with low-bond dimension quantum tensor networks. In: 33st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN, vol 2025

Gray J (2018) quimb: a python library for quantum information and many-body calculations. J Open Source Softw 3(29):819. https://doi.org/10.21105/joss.00819

Huang H-Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H, McClean JR (2021) Power of data in quantum machine learning. Nat Commun. https://doi.org/10.1038/s41467-021-22539-9

Huber PJ (1964) Robust estimation of a location parameter. Ann Math Stat 35(1):73–101. https://doi.org/10.1214/aoms/1177703732

Huggins W, Patil P, Mitchell B, Whaley KB, Stoudenmire EM (2019) Towards quantum machine learning with tensor networks. Quantum Sci Technol 4(2):024001

Huggins W, Patil P, Mitchell B, Whaley KB, Stoudenmire EM (2019) Towards quantum machine learning with tensor networks. Quantum Sci Technol 4(2):024001

Hutter F, Hoos H, Leyton-Brown K (2014) An efficient approach for assessing hyperparameter importance. In: Xing EP, Jebara T (eds) Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol 32, pp 754–762. PMLR, Bejing, China. https://proceedings.mlr.press/v32/hutter14.html

Jia S, Tang Y, Sun J, Ding Q (2022) Data-driven active flutter control of airfoil with input constraints based on adaptive dynamic programming method. J Vib Control 28(13–14):1804–1817. https://doi.org/10.1177/10775463211001182

Jobst B, Shen K, Riofrío CA, Shishenina E, Pollmann F (2024) Efficient MPS representations and quantum circuits from the Fourier modes of classical image data. Quantum 8:1544. https://doi.org/10.22331/q-2024-12-03-1544

Kingma DP, Ba J (2017) Adam: A Method for Stochastic Optimization

Liu D, Ran S-J, Wittek P, Peng C, García RB, Su G, Lewenstein M (2019) Machine learning by unitary tensor network of hierarchical tree structure. New J Phys 21(7):073059

Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Katz R, Himmelfarb J, Bansal N, Lee S-I (2020) From local explanations to global understanding with explainable AI for trees. Nat Mach Intell 2(1):2522–5839

Murg V, Verstraete F, Schneider R, Nagy PR, Legeza O (2015) Tree tensor network state with variable tensor order: An efficient multireference method for strongly correlated systems. J Chem Theory Comput 11(3):1027–1036. https://doi.org/10.1021/ct501187j

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830

Pérez-Salinas A, Cervera-Lierta A, Gil-Fuster E, Latorre JI (2020) Data re-uploading for a universal quantum classifier. Quantum 4:226. https://doi.org/10.22331/q-2020-02-06-226

Quero D, Vuillemin P, Poussot-Vassal C (2019) A generalized state-space aeroservoelastic model based on tangential interpolation. Aerospace 6(1):9. https://doi.org/10.3390/aerospace6010009

Ran S-J (2020) Encoding of matrix product states into quantum circuits of one- and two-qubit gates. Phys Rev A 101:032310. https://doi.org/10.1103/PhysRevA.101.032310

Rauseo M, Vahdati M, Zhao F (2021) Machine learning based sensitivity analysis of aeroelastic stability parameters in a compressor cascade. Int J Turbomach Propuls Power. https://doi.org/10.3390/ijtpp6030039

Reyes J, Stoudenmire M (2020) A Multi-Scale Tensor Network Architecture for Classification and Regression. arXiv:2001.08286

Rieser H-M, Köster F, Raulf AP (2023) Tensor networks for quantum machine learning. Proc R Soc Lond A Math Phys Eng Sci. https://doi.org/10.1098/rspa.2023.0218

Sabater C, Stürmer P, Bekemeyer P (2022) Fast predictions of aircraft aerodynamics using deep-learning techniques. AIAA J 60(9):5249–5261. https://doi.org/10.2514/1.j061234

Schuld M, Bocharov A, Svore KM, Wiebe N (2020) Circuit-centric quantum classifiers. Phys Rev A 101:032308. https://doi.org/10.1103/PhysRevA.101.032308

Schuld M, Sweke R, Meyer JJ (2021) Effect of data encoding on the expressive power of variational quantum-machine-learning models. Phys Rev A 103(3):032430. https://doi.org/10.1103/PhysRevA.103.032430

Shen K, Jobst B, Shishenina E, Pollmann F (2024) Classification of the Fashion-MNIST Dataset on a Quantum Computer. arXiv:2403.02405

Sierra G, Martin-Delgado MA (1998) The Density Matrix Renormalization Group, Quantum Groups and Conformal Field Theory

Stoudenmire EM, Schwab DJ (2016) Supervised learning with tensor networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS'16, pp 4806–4814. Curran Associates Inc., Red Hook, NY, USA

Tewari A (2015) Aeroservoelasticity: Modeling and control. Control Eng. https://doi.org/10.1007/978-1-4939-2368-7

Vidal G (2007) Entanglement renormalization. Phys Rev Lett 99:220405. https://doi.org/10.1103/PhysRevLett.99.220405

Wall ML, Titum P, Quiroz G, Foss-Feig M, Hazzard KR (2022) Tensor-network discriminator architecture for classification of quantum data on quantum computers. Phys Rev A 105(6):062439

Watanabe S, Bansal A, Hutter F (2023) PED-ANOVA: Efficiently Quantifying Hyperparameter Importance in Arbitrary Subspaces . arXiv:2304.10255

Wiersema R, Lewis D, Wierichs D, Carrasquilla J, Killoran N (2024) Here comes the SU(N): multivariate quantum gates and gradients. Quantum 8:1275. https://doi.org/10.22331/q-2024-03-07-1275

Yan F, Iliyasu AM, Venegas-Andraca SE (2015) A survey of quantum image representations. Quant Inf Process 15(1):1–35. https://doi.org/10.1007/s11128-015-1195-6

Zahn R, Breitsamter C (2022) Airfoil buffet aerodynamics at plunge and pitch excitation based on long short-term memory neural network prediction. CEAS Aeronaut J 13:45–55. https://doi.org/10.1007/s13272-021-00550-6