Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Entanglement is Half the Story: Post-Selection vs. Partial Traces

**Gustav J L Jäger**

gustav.jaeger@dlr.de

German Aerospace Center

**Krzysztof Bieniasz**

German Aerospace Center

**Martin B Plenio**

University of Ulm

**Hans-Martin Rieser**

German Aerospace Center

**Research Article**

**Keywords:** Quantum Machine Learning, Tensor Networks, Hyperparameters, Post-Selection, Quantum Classical Hybrid

**Posted Date:** May 4th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-9561473/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# Entanglement is Half the Story: Post-Selection vs. Partial Traces

Gustav J L Jäger[1*], Krzysztof Bieniasz[1], Martin B Plenio[2], Hans-Martin Rieser[1]

[1]Institut für KI-Sicherheit, Deutsches Zentrum für Luft- und Raumfahrt e.V., Wilhelm-Runge-Straße 10, Ulm, 89081, Germany.
[2]Institut für Theoretische Physik and IQST, Universität Ulm, Albert-Einstein-Allee 11, Ulm, 89081, Germany.

*Corresponding author(s). E-mail(s): gustav.jaeger@dlr.de;

**Abstract**

While tensor networks have their traditional application in simulating quantum systems, in the recent decade they have gathered interest as machine learning models. We combine the experience from both fields and derive how quantum constraints placed on a tensor network manifest a change in capabilities. To this end, we employ a method of inference of classical tensor networks on a quantum computer to define a hybrid architecture. This hybrid tensor network is a practical unified framework for it's classical and quantum tensor network edge cases. We identify post-selection as the important property on which this interpolation hinges. The amount of post-selection corresponds to the level to which quantum constraints are enforced on the tensor network. On this basis, we propose a new hyperparameter which controls the transition between the hybrid and the quantum tensor network. In the comparison of classical and quantum tensor networks it complements the bond dimension. Quantum machine learning is improved by using the hyperparameter to allocate the practically limited post-selection to the quantum model in a trainable manner.

**Keywords:** Quantum Machine Learning, Tensor Networks, Hyperparameters, Post-Selection, Quantum Classical Hybrid

# 1 Introduction

With machine learning (ML) becoming a vital field of the sciences, the field of quantum computing has spawned the interest in quantum machine learning (QML). In ML, non-linearities are vital for attaining good results, e.g. prominent activation functions like the sigmoid in NN [1]. On the other hand, QML struggles with imparting non-linearities due to the constraints on quantum systems and their evolution [2]. However, quantum effects like superposition and entanglement as well as the exponentially growing Hilbert space may enable efficient solutions to classically hard problems [2, 3]. In this work, we will take a step towards the understanding between classical and quantum machine learning by addressing and outlining the differences between the ML architectures of classical and quantum tensor networks (TNs). We outline how these differences can be quantified, proposing a hyperparameter to the TN architecture in addition to the commonly used bond dimension. We identify the amount of post-selection as the core difference between the two architectures. Furthermore, on this basis we present a method to improve the performance of QML methods in a multi-shot context. It takes advantage of post-selecting ancilla qubits to introduce local non-linearities. To this end, let us introduce TNs with their background in quantum physics and their use case as an ML model.

To begin with, let us look at TNs in their traditional quantum background: Over the last decades TNs have emerged and been developed for the description and simulation of quantum systems [4–8] and pose a state of the art method for this application [9–11]. Tensor networks compress the description of a quantum system by representing subsystems and their interactions with tensors and their bonds, respectively [6]. As TNs are classical constructs, they do not necessarily obey the fundamental quantum properties of normalization of the state, hermitian observables, and completely positive trace preserving (CPTP) maps [12, 13]. Any TN that globally corresponds to these constraints is called a quantum tensor network (QTN) [14]. For TNs without loops, these constraints are commonly implemented using orthogonality centers [7]. Such methods also exist for more complex TNs but they restrict the space of available states [15]. Common QTNs include matrix product states (MPS), tree tensor networks (TTN), multi-scale entanglement renormalization ansatz (MERA), and projected entangled pair states (PEPS) [5–8]. In the context of the young field of QML, QTNs are used as ML models [16, 17], architectures for variational circuits [16], encodings and classical dimensionality reduction techniques [18, 19], and initializations via classical pre-training for quantum models [20].

TNs have been applied as classical ML models during the last decade. Dropping quantum constraints, the first architectures were simple methods based on classical tensor networks (CTNs) like MPS, also referred to as tensor trains (TTs) in this context [21, 22]. This has been developed further to more complex architectures [16, 17]. Deeper investigations into properties of TNs in the ML context lead to a significant improvement in understanding the combination of both methods, e.g. the impact of mutual information for ordering inputs [23], and observed grokking [24, 25]. Recent efforts seek to embed TNs into larger classical ML setups [26, 27].

Recently, hybrid quantum-classical TNs (HTNs) have been proposed to combine advantages of TN both in quantum and ML[28]. One main idea of HTNs is to project

the capabilities of quantum computing to a large dimension which currently is not reached by quantum computers or at which quantum computers are still inefficient [19, 28–33]. In [28] more efficient simulations of larger quantum systems were achieved by combining CTNs with a smaller QTN. Others expand the limits of NISQ devices for QML tasks by using a CTN [30, 33]. One motivation is to increase the size of datasets that can be handled by the hybrid algorithm and compressing the data for QML using classical dimensionality reduction, see [19, 29]. In the end, TNs are envisioned to seamlessly merge ML capabilities of quantum computing with classical approaches [34].

The structure of the work is as follows: In Sec. 2.1 we give background on the lack of non-linearities in quantum systems and the need to introduce them as pre- or post-processing steps. In this context, we provide a toy example to demonstrate how post-selection can be used as a post-processing step. In Sec. 2.2 we analyze the differences in structure of CTNs and QTNs and propose an inference strategy for CTNs on quantum computers based on post-selection. Sec. 2.3 features the definition of a hybrid tensor network architecture that includes both CTNs and QTNs. Post-selection is identified as the core difference between CTNs and QTNs in this hybrid architecture. Sec. 3 outlines some practical choices and assumptions including about the classification problem, the normalization of the output, and the loss function. In Sec.4 we identify the hyperparameter and give the required proof. The numerical results are given in sec. 5. We investigate the behaviour of the hybrid model dependent on the hyperparameter. Lastly, in Sec. 6 we conclude the work and give an outlook.

# 2 Methods

## 2.1 Post-Processing by Post-Selection

One of the limitations of QML is the lack of global non-linearities. With their linear operations, QML models cannot improve global separation of the data, which is needed for classification tasks. Quantum channels can only compress already existing global separation into a smaller dimensional Hilbert space and thus make it accessible for readout, i.e. they address local separation [2]. For improving global separation, QML models need to rely on pre- and post-processing, because these can introduce non-linearities. One common approach is the so-called data reuploading, i.e. the sequential multiple encoding of the same classical data into a quantum circuit. This allows for the introduction of non-linearities at the pre-processing level [2, 35]. However, in the case of quantum data generated by another quantum system the ability to pre-process is limited. In such cases post-processing is needed. In this section, we identify a viable approach to cause global separation based on post-selection. To this end, let us first give more background on quantum operations.

**Separation of States with Quantum Channels**

Commonly in quantum computing, quantum states are represented by normalized complex vectors $|\Psi\rangle$, also known as ket vectors. They are evolved through unitary operators $U$ that perform a rotation in a complex vector space. However, more generally quantum states are represented by positive semidefinite hermitian density matrices

$\rho$ with unit trace. The general quantum operations are given by completely positive trace preserving (CPTP) linear maps [12, 13]: Any quantum channel $\Lambda$ applied to a mixed quantum state $\rho$ may be expressed as a set of Kraus operators $K_i$, s.t.

$$\Lambda[\rho] = \sum_i^r K_i \rho K_i^\dagger, \quad \text{with} \quad \sum_i^r K_i^\dagger K_i = I. \tag{1}$$

The smallest number of Kraus operators needed to describe a channel $\Lambda$ is called its Kraus rank $r$ [12]. Thus, for $r = 1$ the Kraus operators have to be isometries that project the quantum state to a larger or equal space but they can not reduce the Hilbert space, i.e. project the Hilbert space to a smaller one. The maximum rank necessary to obtain any channel acting on $\rho$ is $\dim(\rho)\dim(\Lambda[\rho])$ [12]. Alternatively, a channel can be rewritten using Stinespring's dilation [6, 12, 13] to be expressed as a unitary acting on the system $A$ and a pure ancilla system $B$ as

$$\Lambda[\rho_A] = \mathrm{tr}_B \left( U_{AB}(\rho_A \otimes |0\rangle\langle 0|_B) U_{AB}^\dagger \right). \tag{2}$$

From this, it follows that with sufficiently large ancilla system $B$, it is possible to represent a quantum channel acting on $A$ as a linear operation on the larger system. Such a linear operation is not able to globally separate the quantum states for a classification problem any more than they already are. This is caused by the quantum data processing inequality [13]

$$S(\Lambda[\rho] \,||\, \Lambda[\sigma]) \leq S(\rho||\sigma) \tag{3}$$

which states that any quantum channel may only decrease the relative entropy between two quantum states. A similar statement can be formulated for the cross-entropy. The overall quantum channel may only project the separating information to a smaller subsystem, while the best possible classification is still limited by the encoding and the resulting globally optimal separation. On the full system $AB$ including the ancilla, there exists an optimal split of the Hilbert space into $l$ orthonormal subspaces for a classification of $l$ labels. Then each of these subspaces may be mapped to the computational states $|l\rangle\langle l|$. This quantum map can then be represented in the two manners shown above in Eqs. (1) and (2).

There are two ways to further improve the classification result: Either improve the encoding through classical pre-processing or use classical post-processing. The pre-processing path is directly linked to the question: Which problems are well suited for QML and how do we map them to QML? Improving the encoding leads to better results for one-shot quantum circuits as they make the problem easier for the quantum computer. For the post-processing path the question is: How can we use the obtained results efficiently? This question only arises in a multi-shot context, where a statistical ensemble of the output of the quantum computer is aggregated. It is this latter question we aim to answer by comparing QTNs and CTNs.

Let's take a look at an example, how we might improve the performance through post-processing. A simple post-processing step that is common in quantum computing

is post-selection, which is simply disregarding certain measurement outcomes from the statistic. This is commonly done without loss of generality by disregarding all the shots of the quantum computer for which the post-selected qubits are not measured as $|0\rangle\langle 0|$. Thus, it requires multiple runs on the quantum device to achieve success.

**Example – Post-Selection and Separation of Quantum States**
In this example, post-selection addresses the core problem in quantum computing that for the separation of two states $|a\rangle$ and $|b\rangle$ by a circuit, the limit for the separation is given by the initial overlap $|\langle a|b\rangle|^2$. In Fig. 1 we choose $|a\rangle = |{+}1\rangle$ and $|b\rangle = |1{+}\rangle$. There are no completely orthonormal subspaces to separate our two states because the states have a non-zero overlap. However, the subspace on which the two states overlap can be post-selected out by removing all shots that originate from a $|11\rangle$ input.

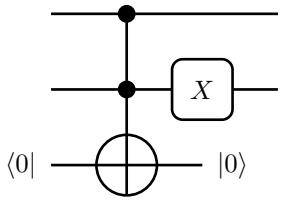



**Fig. 1**: Toffoli gate with post-selection that either separates $|{+}1\rangle$ and $|1{+}\rangle$ with certainty or disqualifies the shot with post-selection. After post-selection $|{+}1\rangle$ is output as $|00\rangle$ and $|1{+}\rangle$ as $|11\rangle$.

As a result, we obtain that a high level of required post-selection indicates that either the dataset as a whole may be unsuitable to QML because they contain a lot of overlapping information, the circuit ansatz is not suitable, or the encoding translates an insufficient amount of separation between the Hilbert subspaces associated with the different classes, i.e. the problem was not already solved by the classical preprocessing and the QML model would simply project down the information from the inaccessibly large Hilbert space. In the case of directly obtained quantum data without a choice of encoding, post-selection can improve the classification accuracy in a multi-shot environment substantially by identifying and discarding runs that contain overlap between classes. A clear benefit is that the model removes potential misclassifications by post-selecting them out.

## 2.2 Moving between Classical and Quantum Tensor Networks

In this section we review the core differences of the architectures of CTNs and QTNs with respect to their application in ML and QML. The fundamental difference is that QTNs equate to a quantum channel and thus any QTN is a CPTP map, see Eqs. (1) and (2). Additionally, quantum systems can be represented by QTNs with large bond dimensions for which it is unfeasible to contract or even store them classically, due to the exponentially growing Hilbert space. Thus, we observe two qualities that have opposite effects on the expressivity of QTNs as compared to CTNs. While QTNs can benefit from quantum simulation by overcoming the limitation of the bond

dimension inherent to all TNs, their expressivity is nonetheless limited by the quantum constraints. There has been a lot of work to address optimization given such constraints, with prominent methods derived from the field of Riemannian optimization [22, 36]. However, in the context of machine learning it is an open question how the introduction of these constraints affects the learning capabilities, which is precisely the question we wish to study here, in order to to gain insight into the potential of QML.

**Current Mapping between CTNs and QTNs**

The common approach to map between QTNs and CTNs architectures is sketched in Fig. 2, see also [16, 20]. Quantum constraints are commonly enforced by allowing partial traces and letting each tensor be isometric, s.t. a smaller input is mapped to a larger output according to the directions in Fig. 2. Thus, there are no isometries that project a larger input to a smaller output as that would require post-selection. Then, we may convert any CTN to a QTN by restricting the respective singular values to one and converting isometries that project down into unitaries that are partially traced. Any QTN architecture may be converted to a CTN by lifting this restriction on the singular values and turning partial traces into post-selection. In this process new degrees of freedom emerge in either direction, one being the singular values and the other presenting when converting the isometries into unitaries with partial traces.

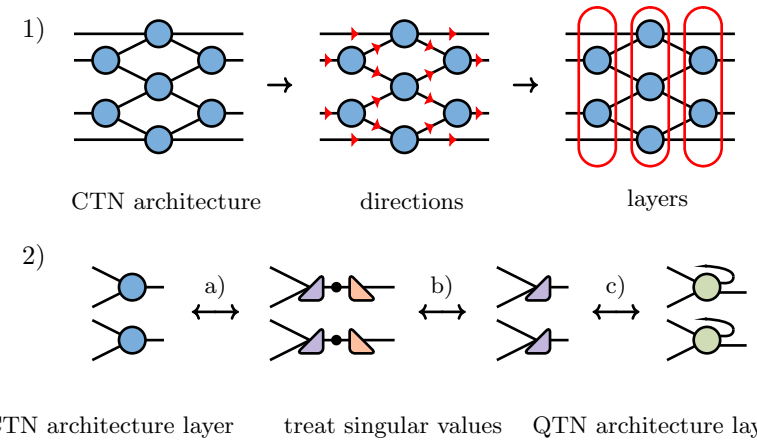



**Fig. 2**: Process for the transition from a CTN architecture to a QTN architecture. 1) General approach of defining the layers is given, which can be applied to both QTNs and CTNs. 2) Transformation of the tensors in a layer necessary for the transition from either CTN to QTN or vice versa. The individual steps are: a) SVD or contraction, b) impose or lift restrictions on singular values, and c) convert between isometries (with post-selection) and unitaries (with partial traces).

**Inference of CTNs on Quantum Computers**

However, QTNs and CTNs can both be executed on universal quantum computers if post-selection is allowed. Generally for any TN, the translation into a quantum circuit

requires the grouping of the open indices into inputs and outputs, like the directions in Fig. 2. Thus, the TN can be reshaped into a matrix operation. This matrix operation is then formatted to conform to quantum constraints and subsequently transpiled into a quantum circuit. For QTNs this procedure is trivial. CTNs however, require post-selecting and rescaling of the quantum circuit results. This process is demonstrated in Fig. 3 for a $2 \times 2$ matrix. Similar methods are implicitly found in quantum computing approaches to partial differential equations [37, 38] or quantum signal processing [39], where non-unitary linear maps are applied on quantum computers [40]. The approach from Fig. 3 can easily be extended to larger matrices by including multi-controlled rotations (MCRY gates). Each MCRY then addresses one of the singular values obtained by the singular value decomposition of the original matrix operation, see Fig. 3. In a qubit efficient approach allowing mid-circuit measurements this process requires only a single ancilla qubit. Furhtermore, the approach can also be easily implemented using deferred measurements on the ancilla. Although the required ancilla count then jumps to at most as many ancillas as singular values.

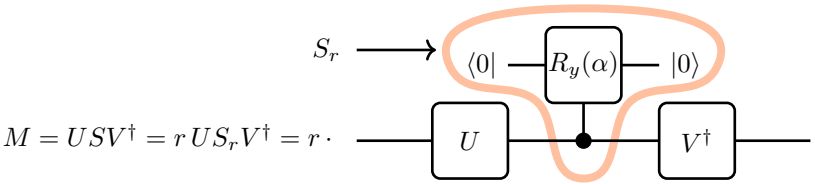



**Fig. 3**: Inference of an arbitrary $2 \times 2$ matrix on a single qubit through post-selection. The singular values are rescaled by the largest singular value $r$, such that the largest entry is 1. The other singular values $d = \cos(\alpha/2)$ is implemented using a controlled rotation with angle $a$.

Thus, one can theoretically run inference of any tensor network on a quantum computer, i.e. any QTN or CTN. However, a practical limitation is quickly found in the amount of post-selection that is possible on a quantum computer. Quantum resources are costly, and the number of singular values and thus multi-controlled rotations grow exponentially with the number of qubits. However, more efficient approaches may be derived from common implementations of diagonal unitaries [40]. Practically, the amount of possible post-selection may be given by the number of total available shots and a minimal number of shots necessary for obtaining a statistically meaningful result. This lower bound may be linked to the expected fidelity of the output.

## 2.3 Tensor Network Architectures: Classical vs. Quantum

Given the comparison of QTNs and CTNs in Sec. 2.2, we observe that the core differences lie in bond dimensions, the singular values, and how the Hilbert space is reduced (partial trace or post-selection). Furthermore, we have also seen the protocol that allows for the inference of CTNs on quantum computers and thus their representation as QTNs. Based on these observations we now define a common architecture that allows for a smooth transition between CTNs and QTNs.

**Defining a Hybrid Tensor Network**

To this end, we have to address some fundamental choices so that QTNs and CTNs can be compared. From Eqs. (1) and (2) and the correspondence of a QTN to CPTP map, it follows that the QTN can be represented as a TN that is contracted with its complex conjugate. On the other hand, the first CTNs to be used in ML were of an MPS like architecture [21, 22] and contained only real valued tensors. Therefore, to allow a genuine comparison, we widen the CTNs to also contain complex values and similarly allow the contraction with it's complex conjugate. Complex valued classical ML models are not unprecedented given the field of CVNNs [41–43]. Furthermore, because QTNs are maps it is valid to assume a grouping into input legs, i.e. bonds, and output legs analogous to the grouping in Fig. 2.

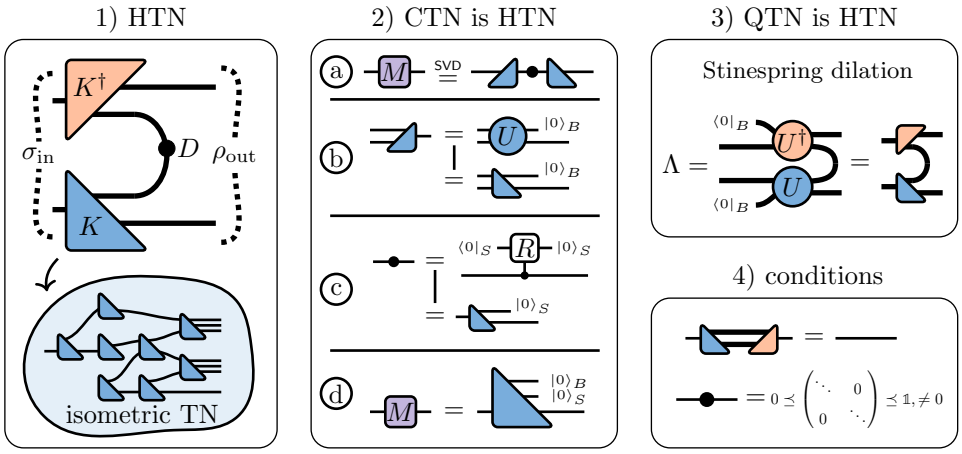



**Fig. 4**: Overview of the proposed HTN with 1) its structure, 2a)-d) rewriting of a CTN into a HTN format, 3) rewriting of a QTN into HTN format, and 4) TN conventions of triangles for isometries and dots for diagonal matrices.

We can now start by introducing the HTN and later defining QTNs and CTNs as special cases.

**Definition 1** Any TN is an HTN if it can be split into three parts: A TN where each tensor is isometric, its conjugate, and a set of real diagonal matrices $0 \preceq D_j \preceq I$ with $D_j \neq 0$, which we will call the reduction operators, which connect the first TN with its conjugate. The isometric TN may only project the input to a larger or equal Hilbert space, i.e. the direction in which it is isometric is relevant. Projection from a larger to a smaller Hilbert space is solely achieved via the reduction operators. With out loss of generality we will aggregate all reduction operators into one $D = \otimes_j D_j$ via tensor product. The parts of the HTN are portrayed in Fig. 4.1.

The outlined architecture in Fig. 4.1 is able to represent both QTNs and CTNs, see Props. 1 and 2. Thus, it serves as a well suited definition for the hybrid of QTNs

and CTNs. However, it differs from the understanding of HTNs in previous works, see [19, 28–34]. While they use a combined architecture whose parts are individually either QTNs or CTNs, we define our HTNs as an architecture that can both take on the role of a CTN or QTN depending on the task. This definition has the core aim of highlighting the differences between CTNs and QTNs.

**Proposition 1** *QTNs are HTNs with identities as diagonal tensors, i.e., any QTN can be represented as an HTN with all the reduction operators as identities, and any HTN with identities can be viewed as a QTN.*

Proof: Using the Stinespring dilation Eq. (2) any quantum channel may be represented by a unitary acting on the input and an ancilla system $|0\rangle\langle 0|_B$ which is traced out at the end. Absorbing, i.e. contracting, the ancilla into this unitary yields the isometry required for the HTN, see Fig. 4.3. Then, we identify the partial trace lead to reduction operators $D$ that are identities. Thus, any quantum channel can be rewritten in accordance with our definition of an HTN. Furthermore, an isometry followed by partial traces is a CPTP map, as is evident from the Stinespring dilation. Thus, any HTN with identities for the reduction operators is always a QTN.

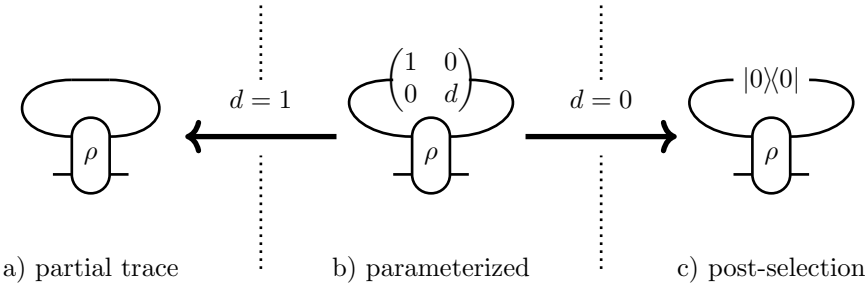
**Proposition 2** *CTNs are HTNs up to a global scaling factor if the reduction operators are one-hot matrices, i.e. they are zero except one diagonal element that is set to one. (This comes with the caveat that only the diagonal entries of the output density matrix are evaluated, which is equivalent to a measurement in the computational basis. This is because CTNs are usually not contracted with their complex conjugate.)*

Proof: The arguments layed out here are visualized in the TN language in Fig. 4.2. Firstly, we may take the singular value decomposition of all tensors in the CTN between the bonds in the input and output direction, see Fig. 4.2a. To this end, each bond has to first be assigned this direction, which might be obtained from the contraction order, see Fig. 2. This results in two isometries and a tensor for the singular values in between. Any isometry with a larger input dimension compared to output dimension is isometric in the wrong direction, i.e. it projects down from a larger space to a smaller Hilbert space. The goal is to isolate the Hilbert space reducing operations and implement them via the reduction operators $D$. To this end, any such isometries can be rewritten by a unitary which is contracted by a vector $|\varphi\rangle_B$ often chosen to be $|0\rangle_B$, see. Fig. 4.2b. This process is commonly known as post-selection in quantum computing. The obtained reduction operators $D_j = |0\rangle\langle 0|$ are one-hot.

All that is left to address are the singular values. They may be implemented using the strategy identified in Sec. 2.2. To this end, they are rescaled s.t. any non-zero diagonal matrix of singular values has it's largest entry as 1, which results in the global scaling factor mentioned in Prop. 2. We disregard the global scaling factor because we will later assume a normalization of the output of the HTN, see Sec. 3. With a controlled rotation on ancilla qubits and subsequent post-selection of these ancilla qubits, one can implement the singular values using a unitary operation followed

by a post-selection, see Fig. 4.2c. Again the resulting reduction operators are one-hot. Subsequently, any CTN can be written, although cumbersome, as an HTN, see Fig. 4.2d.

**Parameterized HTN and CTN/QTN Edge Cases**
Similar ideas of parameterized diagonal matrices are found in rank regularization for tensor networks [44]. However, here they are solely placed on the Hilbert space reducing bonds as reduction operators. Fig. 5 portrays how the reduction operators have the edge cases post-selection and partial trace that are associated with CTNs and QTNs respectively.




a) partial trace b) parameterized c) post-selection

**Fig. 5**: Different approaches to reduce the density operator $\rho$ of a quantum state to a Hilbert space of lower dimension. The parameterized approach in b) for a $2 \times 2$ reduction operator has two edge cases: a) the partial trace with $d = 1$, and c) post-selection with $d = 0$.

# 3 A HTN ML Model

In Sec. 2.1 we saw an example of how post-selection can be applied as a post-processing step in QML of a classification task. In this section, we introduce a mathematical description of the HTN model similar to CTN machine learning models [21]. It can then be used to define and explain the hyperparameters in Sec. 4. Additionally to build a complete ML pipeline, we address the loss function and encoding strategy of classical data.

To this end, let's move from a tensor network to a general description of the map that corresponds to an HTN. It can be given as

$$\Lambda[\sigma_i]_A = \mathrm{tr}_B(U\sigma_i U^\dagger(D_B \otimes I_A)). \tag{4}$$

where the $U$ corresponds to the isometric TN found in Figs 4.1. We choose $U$ to be unitary by moving the ancilla system necessary for isometries $U_{\text{iso}} = U(|0\rangle \otimes I)$ into the encoded data $\sigma_i$. Thus, from here on out $\sigma_i$ contains not only the data but also a sufficiently large system at $|0\rangle$. $D_B$ is the aggregate tensor product of the reduction operators. Overall, the system is divided into subsystems $A$ and $B$, where $A$ is the system of the output and on $B$ is completely removed by the reduction operators.

We apply this map to a classification task and assume that the labels are encoded as pure target states,

$$\tau_i = |\tau_i\rangle\langle\tau_i| \,. \tag{5}$$

Without loss of generality, we can identify these pure states as the computational basis with the label $l$ encoded as $|l\rangle\langle l|$. Due to the post-selection, the trace of the density operator output by the channel in Eq. (4) is effectively no longer equal to 1. To compare the output state with the target states, an amplification of the trace is necessary. In CTNs, it is common to find the diagonal matrices obtained by SVD to have entries larger than 1, which corresponds to a simple multiplying factor equally applied to each output state. However instead of treating each state equally, better losses are achieved if each output state is individually normalized. To this end, we divide each output state $\rho_i = \Lambda[\sigma_i]$ by it's trace,

$$\mathcal{N}[\rho_i] = \frac{\rho_i}{\mathrm{tr}(\rho_i)}. \tag{6}$$

This step also allows us to disregard the global scaling factors of CTNs.

For non invertible matrices like density operators of pure quantum states, the matrix logarithm in Eq. (8) is ill-defined. To circumvent this issue, we use a depolarizing channel $\Delta_\lambda$. The quantum state is set to the maximally mixed state with a small probability $\lambda$ that we choose sufficiently large to ensure numerical stability. The depolarizing channel [13] for a quantum state $\rho$ with dimension $d$ is defined as

$$\Delta_\lambda[\rho] = (1-\lambda)\rho + \lambda\frac{I}{d}. \tag{7}$$

With the cross entropy loss [1, 45] we can now obtain our loss function,

$$L = \frac{1}{N}\sum_{i=1}^{N} -\mathrm{tr}\left(\tau_i \log \Delta_\lambda \circ \mathcal{N} \circ \Lambda[\sigma_i]\right). \tag{8}$$

Similarly to CTN ML models [21] the MSE loss might be a more practical choice.

# 4 Proposed Hyperparameter

One prominent property of TNs is the bond dimension and in the context of machine learning it is an obvious hyperparameter. Because it can be adapted during training [21, 46], the hyperparamter is typically the maximum bond dimension to which larger bonds are truncated. For CTNs and generally for QTNs the bond dimension is monotonous in the optimized loss, such that a larger bond dimension results in a better or equal loss. However, it is limited classically by storage constraints, while quantum systems have states with arbitrary large bond dimension. Due to the current state of hardware, we use QTNs as a stand in for quantum systems and are thus also limited in the bond dimension. In a sense the bond dimension is a resource which favours quantum systems. However, given a fixed bond dimension, previous results

have shown that the introduced CPTP constraints can significantly impact the performance negatively [47]. This gap in the comparison of CTNs and QTNs is at the core of this work. To this end, we introduce a new hyperparameter, that controls this introduction of quantum constraints on the previously defined HTN. At its heart, this hyperparameter allows for the transition between strict enforcement and complete relaxation of the quantum constraints on the hybrid architecture of the HTN. However, it does not allow for the full control between QTNs and CTNs directly, which would require an additional hyperparameter possibly linked to the purity of the output. Instead it allows us to ascertain how well the QTN performs in the comparison to the overarching HTN architecture.

Given Eq. (8) and $\lambda = 0$, we want to define an $h$-dependent loss function $L_h$, such that

$$\begin{aligned} L_0 &= \frac{1}{N}\sum_{i=1}^{N} -\operatorname{tr}\left(\tau_i \log \Delta_\lambda \circ \mathcal{N} \circ \Lambda[\sigma_i]\right), \quad \text{and} \\ L_1 &= \frac{1}{N}\sum_{i=1}^{N} -\operatorname{tr}\left(\tau_i \log \Delta_\lambda \circ \Lambda[\sigma_i]\right). \end{aligned} \tag{9}$$

where $h \in [0,1]$ controls whether the normalization is applied or not. A practical choice is to introduce a threshold $t \in [0,1]$ on how much normalization is applied

$$\mathcal{N}_t[\rho_i] = \frac{\rho_i}{\max(\operatorname{tr}(\rho_i), t)}. \tag{10}$$

This practical approach is oriented towards limiting the percentage of discarded shots. Alternatively, the hyperparameter $w \in [0,1]$ can be motivated from information theory and limits the normalization to

$$\mathcal{N}_w[\rho_i] = \frac{\rho_i}{\operatorname{tr}(\rho_i)^{(1-w)}}. \tag{11}$$

For $\lambda = 0$, i.e. no depolarizing channel, the information weighting emerges by using the properties of the logarithm on $L_w$:

$$L_w = \frac{1}{N}\sum_{i=1}^{N} -\operatorname{tr}\left(\tau_i \log \Lambda[\sigma_i]\right) - (1-w)\frac{1}{N}\sum_{i=1}^{N} -\log\operatorname{tr}\left(\Lambda[\sigma_i]\right). \tag{12}$$

We observe that for $L_{h=1}$ the normalization disappears. We show that $L_1$ is minimized by setting all reduction operators equal to identities. Thus, $L_1$ is minimized by avoiding post-selection completely and using partial traces instead. This equates to strictly adhering to the trace preserving part of the CPTP constraint and thus it results in quantum channels, recall Fig.4. Because $h$ transitions between $L_0$ and $L_1$ it is a suitable hyperparameter to control the permitted amount of post-selection.

While for $L_0$ post-selection is allowed, it is not enforced in the strictest sense as $L_0$ is not necessarily minimized by setting all reduction operators to one-hot diagonal

matrices. Thus, the hyperparameter transitions between a QTN and HTN, but not a CTN.

**Proposition 3** *Given the loss $L_1$ in Eq. (9), without affecting the minimized loss one can assume that all diagonal matrices of the HTN architecture in Fig. 6 can be set to the identity.*

Proof: For a single instance of the dataset $i$ with the $\sigma_i$ as the encoded data combined with a sufficiently large ancilla system, we may write the output $\rho_{i,D}$ of the HTN according to Eq. (4) as

$$\rho_{i,D} = \mathrm{tr}_B \left( U \sigma_i U^\dagger (D_B \otimes I) \right), \tag{13}$$

while the QTN output is

$$\rho_{i,I} = \mathrm{tr}_B \left( U \sigma_i U^\dagger \right). \tag{14}$$

Then

$$\rho_{i,I} - \rho_{i,D} = \mathrm{tr}_B \left[ U \sigma_i U^\dagger ((I - D_B) \otimes I) \right] \succeq 0. \tag{15}$$

Since the depolarizing channel preserves order and the logarithm is operator monotone on positive semidefinite matrices [48–50],

$$-\mathrm{tr}[\tau_i \log \Delta_\lambda(\rho_{i,I})] \leq -\mathrm{tr}[\tau_i \log \Delta_\lambda(\rho_{i,D})]. \tag{16}$$

Thus for every (U,D), the choice $D = I$ is no worse, proving the optimum of $L_1$ can be taken at the QTN edge. The proof for the MSE loss is found in App. A.

Furthermore, we can make the following argument:

**Proposition 4** *The losses $L_w$ and $L_t$ change monotonously with the hyperparameters $w$ and $t$ respectively.*

Proof: For $\mathcal{N}_t$

$$t_1 < t_2 \Rightarrow \max(\mathrm{tr}[\rho_{i,D}], t_1) \leq \max(\mathrm{tr}[\rho_{i,D}], t_2) \Rightarrow \mathcal{N}_{t_1} \succeq \mathcal{N}_{t_2}. \tag{17}$$

Again invoking the order preserving of the depolarizing channel and the operator monotone property of the logarithm, we obtain

$$-\mathrm{tr}\left(\tau_i \log \Delta_\lambda \circ \mathcal{N}_{t_1} \circ \Lambda[\sigma_i]\right) \leq -\mathrm{tr}\left(\tau_i \log \Delta_\lambda \circ \mathcal{N}_{t_2} \circ \Lambda[\sigma_i]\right). \tag{18}$$

Thus, the loss $L_t$ is monotone with the hyperparameter $t$. Similarly for $\mathcal{N}_w$

$$w_1 < w_2 \Rightarrow (\mathrm{tr}[\rho_{i,D}])^{w_1} > (\mathrm{tr}[\rho_{i,D}])^{w_2} \Rightarrow \mathcal{N}_{w_1} \succeq \mathcal{N}_{w_2}. \tag{19}$$

Like the above result, $L_w$ is monotone with $w$.

# 5 Numerical Results

To investigate the HTN from Def. 1 we utilize a simple MPS based architecture. While more complex TN architectures offer a wider range of applications, MPS is still a common choice for investigating new TN methods or modifications of the architecture. The MPS based HTN is portrayed in Fig. 6.1. It allows us to investigate the effect of post-selection on our system.

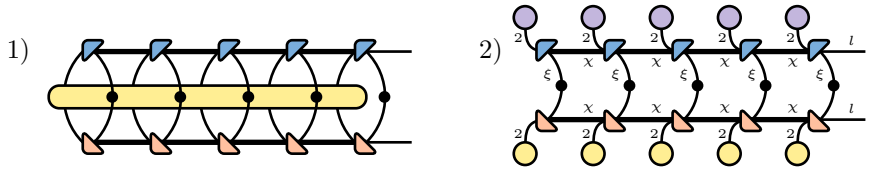
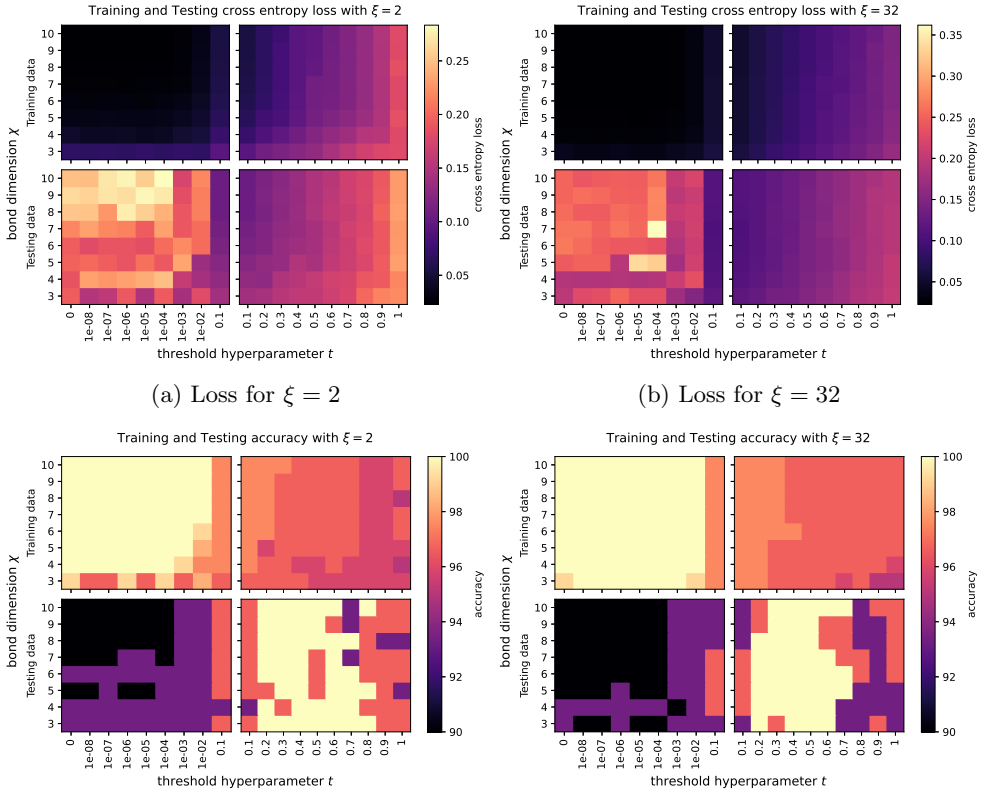



**Fig. 6**: ML model for an MPS-inspired HTN. 1) is assuming an arbitrary encoding of the data into the yellow MPO. 2) show the rotational encoding into the purple and yellow product states. The blue and orange isometric tensors are their respective complex conjugates at each site. The black dots are the reduction operators, i.e. real diagonal matrices $0 \preceq D \preceq I$ with $D \neq 0$ and dimension $\xi$. Contracting all bonds leaves the output state on the open legs on the left.

For the numerical experiments, we choose rotational encoding, see Fig. 6.2. The complexity of the HTN is overall quartic in the bond dimension, while classical MPS inspired ML models are quadratic, see [21]. The product state encoding can help address scaling issues, which is why they are commonly employed for CTN based ML models. In this section, we aim to demonstrate how the previous findings might be applied. Because we focus specifically on the post-processing step, we do not survey other encoding schemes. Similar to [21] we use environment caching during the optimization sweeps, which for the HTN scales quadratic with the bond dimension of the channel, the bond dimension of the encoding, and linear with the size of training data. or a more detailed guide on environment caching refer to [51]. Furthermore, like [21] we use an initialization of the tensor network using the data. We tested the initializations stability and for the extreme cases of $\chi$, $\xi$, and $t$. We found the loss of our initialization to be within 4% of the smallest loss of 200 random intializations and within 3% of their average.




(a) Loss for $\xi = 2$

(b) Loss for $\xi = 32$

(c) Accuracy for $\xi = 2$

(d) Accuracy for $\xi = 32$

**Fig. 7**: Comparison of $\chi$ and $t$ for the reduction operator dimension $\xi = 2$ and $\xi = 32$. For each subplot the top row is the training data and the bottom row for the testing data. On the left is the logarithmic scale for small thresholds, surveying the different orders of magnitude of the maximally allowed normalization, while on the right is a linear scale that transitions the HTN to a QTN.

For the plots in Fig. 7 the Iris dataset[52, 53] was used. The HTN in Fig. 4.2 has the following hyperparameters besides our newly proposed: the bond dimension $\chi$ connecting the isometric tensors, the bond dimension $\xi$ of the reduction operators. If $\xi = 2\chi$, the HTN can express any unnormalized MPS so precisely the architecture pioneered in [21], while setting $\xi = 2$ reduces the number of required ancillas. The small models are trained with 20 two-site sweeps, where at each step of the sweep the local tensor is close to fully optimized using the ADAM optimizer.

For both $\xi = 2$ and $\xi = 32$, the impact of our newly introduced hyperparameter that manages the level of allowed post-selection is larger than that of the bond dimension $\chi$. For the model with $\xi = 32$ that may emulate the respective CTN architecture completely, we observe that the training loss is even invariate with $\chi$. For the training loss, we observe the monotonicity of the loss over the hyperparameter derived in

Prop. 4. Thus, the hyperparameter can serve as a natural method to limit the amount of post-selection permitted in a quantum model, while the corresponding loss function allocates post-selection efficiently. For the testing data, we observe overfitting for small $t$, i.e. for a lot of allowed post-selection as can be seen in from the small training loss and large testing loss in Figs. 7a and 7b. This overfitting causes a collapse in the accuracy of the obtained models, see Figs. 7c and 7d. The emergence of overfitting at small $t$ is due to the fact that a smaller $t$ corresponds to a larger expressivity of the HTN. The overfitting appears slightly stronger for larger bond dimensions $t$. Furthermore, much of the impact and observed behaviour of post-selection is already obtained for small $\xi = 2$, which makes it practical for QML models as few ancillas are needed.

We probe the model capabilities on a dataset with more features a simple binary classification of the MNIST dataset that is rescaled to $7 \times 7$. Although, the model from Fig. 6 can not accurately represent quantum channels for a larger number of sites due to limited bond dimensions. In our test case using $h = 1$, i.e. no post-selection, we ran into a barren plateau [54] with the model returning a fully mixed state for data derived and random initializations. This amounts to random guessing of the output which persists during training due to a vanishing gradient. Additionally to the effects of the curse of dimensionality the machine epsilon increase because of numerical accuracy of the environment caching. For $h = 0$ the final testing accuracy was 99.68%. The other model settings were chosen to be $\chi = 10$ and $\xi = 40$ for both cases of $h$. Thus, the barren plateau is counteracted by selecting subspaces of the Hilbert space that have non-zero gradients.

# 6 Conclusion and Outlook

In this work, we identified the core differences between classical and quantum tensor networks. On the basis of a procedure that allows inference of CTNs on quantum computers, we derived a unified framework that allows for a smooth transition between CTNs and QTNs. Next to the already established bond dimension, we found that post-selection is the property that distinguishes between CTNs and QTNs. We presented the ability of post-selection to improve global separation of data in QML. Because it is a limited resource in real systems, we present a hybrid HTN accompanied by a threshold hyperparameter that allocates a given level of post-selection to the HTN by connecting it to the normalization of the post-selected output.

The numerical results underline that the novel application of post-selection with the presented hyperparameter can benefit QML. Post-selection effectively implements a selective or abstaining classifier by discarding low-confidence samples. However, as a trade-off exists with the number of retained shots. The results are for a specific encoding and it is possible, likely even, that better pre-processing can achieve similar results or even surpass this post-processing routine. Finding a good encoding for QML brings its own set of challenges, while this post-selection based approach can be easily implemented. Because post-selection can detect and remove information that worsens the effectiveness of QML models, the question follows, if post-selection can be used to identify better encodings? To this end, further research is needed to link post- with pre-processing in QML.

This novel approach leads to several other open questions for our future work. A hyperparameter that connects the HTN to a CTN is missing to fully connect quantum and classical models, although the transition is on the basis of post-selection, one path to such a quantity might be enforcing the purity of the output before post-selection is applied which leads to a CTN. This would deepen the understanding for which encoded data the CTN or QTN achieves better results, which would help with the question for which data QML is advantageous. Additionally, while the hyperparameter already allows the trainable allocation of post-selection, it is important to understand how it gets distributed across the HTN and how that relates to the data and encoding. This would lead to a better understanding of weak points of a data and encoding pairing and thus inform on the choice of encoding.

# Declarations

## Author Contributions

G.J developed the main conceptual ideas, set up, carried out, and analyzed the experiments, and provided the initial draft. H-M.R supervised and guided the project. K.B imparted guidance and contributed to the final draft. M.P provided guidance towards the simplification of the theory laid out in the initial draft.

## Conflicts of Interest

The authors declare no competing interests.

## Data availability

Data utilised during the current study are available from the corresponding author on reasonable request. The simulations are available from the DLR but restrictions apply to the availability of these and are therefore not publicly available.


## Funding

G.J received funding from the DLR (German Aerospace Center) through the Quantum Fellowship Program. Furthermore, the project was enabled by the DLR Quantencomputing-Initiative and the German federal ministry of Research, Technology and Space; https://qci.dlr.de/projekte/qutenet


# References


[1] Grohs, P., Kutyniok, G.: Mathematical Aspects of Deep Learning. Cambridge University Press. https://doi.org/10.1017/9781009025096

[2] Schuld, M., Petruccione, F.: Machine Learning with Quantum Computers. Springer. https://doi.org/10.1007/978-3-030-83098-4

[3] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press. https://doi.org/10.1017/cbo9780511976667

[4] White, S.R.: Density matrix formulation for quantum renormalization groups **69**(19), 2863–2866 https://doi.org/10.1103/physrevlett.69.2863

[5] Ran, S.-J., Tirrito, E., Peng, C., Chen, X., Tagliacozzo, L., Su, G., Lewenstein, M.: Tensor Network Contractions: Methods and Applications to Quantum Many-Body Systems. Springer. https://doi.org/10.1007/978-3-030-34489-4

[6] Bridgeman, J.C., Chubb, C.T.: Hand-waving and interpretive dance: an introductory course on tensor networks **50**(22), 223001 https://doi.org/10.1088/1751-8121/aa6dc3

[7] Orus, R.: A practical introduction to tensor networks: Matrix product states and projected entangled pair states **349**, 117–158 (2013) https://doi.org/10.1016/j.aop.2014.06.013 1306.2164

[8] Schollwoeck, U.: The density-matrix renormalization group in the age of matrix product states https://doi.org/10.48550/ARXIV.1008.3477

[9] Pan, F., Chen, K., Zhang, P.: Solving the sampling problem of the sycamore quantum circuits https://doi.org/10.48550/ARXIV.2111.03011

[10] Tindall, J., Fishman, M., Stoudenmire, M., Sels, D.: Efficient tensor network simulation of ibm's eagle kicked ising experiment https://doi.org/10.48550/ARXIV.2306.14887

[11] Berezutskii, A., Liu, M., Acharya, A., Ellerbrock, R., Gray, J., Haghshenas, R., He, Z., Khan, A., Kuzmin, V., Lyakh, D., Lykov, D., Mandrà, S., Mansell, C., Melnikov, A., Melnikov, A., Mironov, V., Morozov, D., Neukart, F., Nocera, A., Perlin, M.A., Perelshtein, M., Steinberg, M., Shaydulin, R., Villalonga, B., Pflitsch, M., Pistoia, M., Vinokur, V., Alexeev, Y.: Tensor networks for quantum computing. arXiv. https://doi.org/10.48550/ARXIV.2503.08626

[12] Bengtsson, I., Życzkowski, K.: Geometry of Quantum States: An Introduction to Quantum Entanglement. Cambridge University Press. https://doi.org/10.1017/9781139207010

[13] Wilde, M.M.: Quantum Information Theory. Cambridge University Press. https://doi.org/10.1017/9781316809976

[14] Rieser, H.-M., Köster, F., Raulf, A.P.: Tensor networks for quantum machine learning **479**(2275) https://doi.org/10.1098/rspa.2023.0218

[15] Zaletel, M.P., Pollmann, F.: Isometric tensor network states in two dimensions

**124**(3), 037201 https://doi.org/10.1103/physrevlett.124.037201

[16] Huggins, W., Patil, P., Mitchell, B., Whaley, K.B., Stoudenmire, E.M.: Towards quantum machine learning with tensor networks **4**(2), 024001 https://doi.org/10.1088/2058-9565/aaea94

[17] Stoudenmire, E.M.: Learning relevant features of data with multi-scale tensor networks **3**(3), 034003 https://doi.org/10.1088/2058-9565/aaba1a

[18] Dilip, R., Liu, Y.-J., Smith, A., Pollmann, F.: Data compression for quantum machine learning https://doi.org/10.48550/ARXIV.2204.11170

[19] Hickmann, M.L., Alves, P., Quero, D., Schwenker, F., Rieser, H.-M.: Hybrid quantum tensor networks for aeroelastic applications **7**(2) https://doi.org/10.1007/s42484-025-00327-8

[20] Wall, M.L., Titum, P., Quiroz, G., Foss-Feig, M., Hazzard, K.R.A.: Tensor-network discriminator architecture for classification of quantum data on quantum computers **105**(6), 062439 https://doi.org/10.1103/physreva.105.062439

[21] Stoudenmire, E.M., Schwab, D.J.: Supervised learning with quantum-inspired tensor networks **29**(4799) https://doi.org/10.48550/ARXIV.1605.05775

[22] Novikov, A., Trofimov, M., Oseledets, I.: Exponential machines (2016) https://doi.org/10.48550/ARXIV.1605.03795 1605.03795

[23] Harada, K., Okubo, T., Kawashima, N.: Tensor tree learns hidden relational structures in data to construct generative models **6**(2), 025002 https://doi.org/10.1088/2632-2153/adc2c7

[24] Pomarico, D., Cilli, R., Monaco, A., Bellantuono, L., La Rocca, M., Maggipinto, T., Magnifico, G., Ortega, M.O., Pantaleo, E., Tangaro, S., Stramaglia, S., Bellotti, R., Amoroso, N.: Transfer entropy and O-information to detect grokking in tensor network multi-class classification problems. arXiv. https://doi.org/10.48550/ARXIV.2507.23346

[25] Pomarico, D., Monaco, A., Magnifico, G., Lacalamita, A., Pantaleo, E., Bellantuono, L., Tangaro, S., Maggipinto, T., La Rocca, M., Picardi, E., Amoroso, N., Pesole, G., Stramaglia, S., Bellotti, R.: Grokking as an entanglement transition in tensor network machine learning. arXiv. https://doi.org/10.48550/ARXIV.2503.10483

[26] Tomut, A., Jahromi, S.S., Sarkar, A., Kurt, U., Singh, S., Ishtiaq, F., Muñoz, C., Bajaj, P.S., Elborady, A., Bimbo, G., Alizadeh, M., Montero, D., Martin-Ramiro, P., Ibrahim, M., Alaoui, O.T., Malcolm, J., Mugel, S., Orus, R.: CompactifAI: Extreme Compression of Large Language Models using Quantum-Inspired Tensor Networks. arXiv. https://doi.org/10.48550/ARXIV.2401.14109

[27] Wang, M., Pan, Y., Xu, Z., Li, G., Yang, X., Mandic, D., Cichocki, A.: Tensor Networks Meet Neural Networks: A Survey and Future Perspectives. arXiv. https://doi.org/10.48550/ARXIV.2302.09019

[28] Yuan, X., Sun, J., Liu, J., Zhao, Q., Zhou, Y.: Quantum simulation with hybrid tensor networks **127**(4), 040501 https://doi.org/10.1103/physrevlett.127.040501 arxiv:2007.00958

[29] Chen, S.Y.-C., Huang, C.-M., Hsing, C.-W., Kao, Y.-J.: An end-to-end trainable hybrid classical-quantum classifier **2**(4), 045021 https://doi.org/10.1088/2632-2153/ac104d arxiv:2102.02416

[30] Huang, J., He, W., Zhang, Y., Wu, Y., Wu, B., Yuan, X.: Tensor-network-assisted variational quantum algorithm **108**(5), 052407 https://doi.org/10.1103/physreva.108.052407 arxiv:2212.10421

[31] Xiu, Y., Ye, F., Chen, Z., Liu, Y.: Hybrid tensor networks for fully supervised and semisupervised hyperspectral image classification **16**, 7882–7895 https://doi.org/10.1109/jstars.2023.3308723

[32] Schuhmacher, J., Ballarin, M., Baiardi, A., Magnifico, G., Tacchino, F., Montangero, S., Tavernelli, I.: Hybrid tree tensor networks for quantum simulation **6**(1), 010320 https://doi.org/10.1103/prxquantum.6.010320 arxiv:2404.05784

[33] Harada, H., Suzuki, Y., Yang, B., Tokunaga, Y., Endo, S.: Density matrix representation of hybrid tensor networks for noisy quantum devices **9**, 1823 https://doi.org/10.22331/q-2025-08-07-1823 arxiv:2309.15761v4

[34] Yao, J.: Hybrid tensor networks: The integration of quantum and classical machine learning. In: 2024 2nd International Conference on Computer, Vision and Intelligent Technology (ICCVIT), pp. 1–5. IEEE. https://doi.org/10.1109/iccvit63928.2024.10872520

[35] Cao, Y., Guerreschi, G.G., Aspuru-Guzik, A.: Quantum neuron: an elementary building block for machine learning on quantum computers (2017) https://doi.org/10.48550/ARXIV.1711.11240 1711.11240

[36] Boumal, N.: An Introduction to Optimization on Smooth Manifolds, 1st ed. edn. Cambridge University Press. https://doi.org/10.1017/9781009166164 . Description based on publisher supplied metadata and other sources.

[37] Siegl, P., Reese, G.S., Hashizume, T., Hülst, N.-L., Jaksch, D.: Tensor-Programmable Quantum Circuits for Solving Differential Equations. arXiv. https://doi.org/10.48550/ARXIV.2502.04425

[38] Termanova, A., Melnikov, A., Mamenchikov, E., Belokonev, N., Dolgov, S., Berezutskii, A., Ellerbrock, R., Mansell, C., Perelshtein, M.: Tensor Quantum

Programming. arXiv. https://doi.org/10.48550/ARXIV.2403.13486

[39] Suzuki, Y., Tiang, B.H., Son, J., Ng, N.H.Y., Holmes, Z., Gluza, M.: Double-bracket algorithm for quantum signal processing without post-selection **9**, 1954 https://doi.org/10.22331/q-2025-12-23-1954

[40] Zylberman, J., Nzongani, U., Simonetto, A., Debbasch, F.: Efficient quantum circuits for non-unitary and unitary diagonal operators with space-time-accuracy trade-offs **6**(2), 1–43 https://doi.org/10.1145/3718348

[41] Hirose, A.: Complex-Valued Neural Networks. Springer. https://doi.org/10.1007/978-3-642-27632-3

[42] Lee, C., Hasegawa, H., Gao, S.: Complex-valued neural networks: A comprehensive survey **9**(8), 1406–1426 https://doi.org/10.1109/jas.2022.105743

[43] Barrachina, J.A., Ren, C., Vieillard, G., Morisseau, C., Ovarlez, J.-P.: Theory and Implementation of Complex-Valued Neural Networks. arXiv. https://doi.org/10.48550/ARXIV.2302.08286

[44] Barratt, F., Dborin, J., Wright, L.: Improvements to gradient descent methods for quantum tensor network machine learning https://doi.org/10.48550/ARXIV.2203.03366

[45] Shangnan, Z., Wang, Y.: Quantum cross entropy and maximum likelihood principle (2021) https://doi.org/10.48550/ARXIV.2102.11887 2102.11887

[46] Geng, C., Hu, H.-Y., Zou, Y.: Differentiable programming of isometric tensor networks **3**(1), 015020 https://doi.org/10.1088/2632-2153/ac48a2 arxiv:2110.03898

[47] Jäger, G., Plenio, M.B., Rieser, H.-M.: Quantum tensor network learning with dmrg. In: ESANN 2025 Proceesdings. ESANN 2025, pp. 537–542. Ciaco - i6doc.com. https://doi.org/10.14428/esann/2025.es2025-157

[48] Chansangiam, P.: A survey on operator monotonicity, operator convexity, and operator means **2015**, 1–8 https://doi.org/10.1155/2015/649839

[49] HIAI, F.: Matrix analysis: Matrix monotone functions, matrix means, and majorization **16**(2), 139–246 https://doi.org/10.4036/iis.2010.139

[50] Furuta, T.: Operator monotone functions, a >b > 0 and loga > logb (1), 93–96 https://doi.org/10.7153/jmi-07-08

[51] Le, I.N.M., Sun, S., Mendl, C.B.: Riemannian quantum circuit optimization based on matrix product operators **9**, 1833 https://doi.org/10.22331/q-2025-08-27-1833

[52] Fisher, R.A.: The use of multiple measurements in taxonomic problems **7**(2),

179–188 https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

[53] Anderson, E.: The species problem in iris **23**(3), 457 https://doi.org/10.2307/2394164

[54] McClean, J.R., Boixo, S., Smelyanskiy, V.N., Babbush, R., Neven, H.: Barren plateaus in quantum neural network training landscapes **9**(1) (2018) https://doi.org/10.1038/s41467-018-07090-4 1803.11173

[55] Gray, J.: quimb: A python package for quantum information and many-body calculations **3**(29), 819 https://doi.org/10.21105/joss.00819

# A The Hyperparameter and the MSE loss

With the partial normalization $\mathcal{N}_h$ dependent on the hyperparameter $h$ (either $t$ or $w$) the MSE loss is defined as

$$L_{\mathrm{MSE},h} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(\tau_i - \mathcal{N}_h \circ \Lambda[\sigma_i])^2. \tag{20}$$

Again, for $h = 0$ the normalization is fully employed and thus we find it a suitable loss function for the general HTN architecture. Now we need to show that for $h = 1$ the loss function is optimal for only QTNs. To this end, for any HTN with Eq. 4 we may define a channel that uses the post-selection to give a randomized output. For any post-selected output $\rho_{i,D}$ we instead obtain

$$\rho_{i,R} = \rho_{i,D} + (1 - \mathrm{tr}(\rho_{i,D}))\frac{I}{\dim(\rho_{i,D})}. \tag{21}$$

Then

$$\mathrm{tr}\left((\rho_{i,R} - \tau_i)^2 - (\rho_{i,D} - \tau_i)^2\right) = -\frac{1 - \mathrm{tr}(\rho_{i,D})}{\dim(\rho_{i,D})} < 0. \tag{22}$$

Thus, following that each individual contribution is negative, we are always able to obtain a QTN that performs equal or better than an HTN. Furthermore, equality is only obtained if no state is post-selected, i.e. $p_i$. This case appears when post-selecting out states that are never occupied.

Given the strategy of randomizing instead of post-selecting, it may very well be possible to generalize the above proofs for the cross entropy loss and MSE loss to arbitrary loss functions and tie the proof to properties of distinguishability measures [12] and statements about operator montonicity [48]. Specifically, using the triangle inequality of distinguishability measures [12].