Enhanced QTTN Design: Scalable Quantum Circuits for Arbitrary Qubit Counts

Krishnageetha Karuppasamy[a] and Johnson P. Thomas[b]
Department of Computer Science, Oklahoma State University, Stillwater, U.S.A.

Keywords: Quantum Tree Tensor Network, Variational Quantum Circuits, Fidelity, Entanglement.

Abstract: We explore the design and implementation of Enhanced Quantum Tree Tensor Networks (EQTTNs) for Variational Quantum Circuits. A Quantum Tree Tensor Network (QTTN) offers a hierarchical structure to manage entanglement and optimize quantum operations. The traditional requirement for constructing a QTTN is that the number of qubits (n) must be in the form $n = 2^x$. This paper proposes an EQTTNs design that can accommodate any number of qubits. This flexibility means there are no restrictions on the problem size, allowing for broader applicability and scalability in various quantum computing tasks. We provide a comprehensive analysis of the parameter count required for EQTTNs. Experimental results validate our theoretical model, in terms of fidelity score and entanglement strength.

1 INTRODUCTION

Variational Quantum Algorithms (VQAs) are a class of quantum algorithms designed to solve optimization and Artificial Intelligence (AI) tasks on quantum computers. VQAs have emerged as the leading strategy to obtain quantum advantage. However, implementation challenges remain because of size limitations and errors that are inherent in Noisy Intermediate-Scale Quantum (NISQ) devices.

A major reason behind the success of variational quantum algorithms (Cerezo et al., 2021) is that VQAs minimize the computational burden on NISQ devices. VQAs achieve high performance by running only the part of the algorithm that will result in quantum advantage on a quantum system, while all remaining tasks are outsourced to a classical computer. VQAs are therefore a class of hybrid quantum-classical algorithms in which quantum and classical computational resources are used in combination to solve a task while achieving high performance.

Quantum Artificial Intelligence (QAI) is one of the most promising areas of VQAs, as VQAs provide a robust framework for implementing QAI on NISQ devices, by optimizing Parameterized Quantum Circuits (PQCs/ansatz) using classical methods. However, optimizing (PQCs) using classical methods faces challenges like barren plateaus, noise, and scalability issues. The efficiency and success of variational quantum algorithms in QAI depends on a well-designed ansatz (Nielsen et al. 2011). Ansatz is a tuneable parameterized quantum circuit. To address these limitations and enhance the applicability of variational quantum algorithms in quantum machine learning, it is crucial to explore ansatz design. Tensor tree networks ansatz, which efficiently represent complex quantum states, offer a promising solution.

Tensor networks are factorizations of very large tensors into networks of smaller tensors used to represent and manipulate large, multi-dimensional arrays of data or tensors efficiently. The ansatz takes any quantum state as input, and through manipulation by a unitary matrix, produces an outcome of a quantum state. Given that this quantum state can be thought of as a high-dimensional vector, it aligns well with the structure of tensor networks. Tensor networks are specifically designed to efficiently represent and manipulate such high-dimensional vectors by breaking them down into smaller, interconnected tensors. Tensor networks are therefore an ideal tool for preparing the ansatz in quantum machine learning. By leveraging tensor networks, we

[a] https://orcid.org/0009-0003-5570-7855
[b] https://orcid.org/0000-0002-4698-8072

Karuppasamy, K. and Thomas, J. P.
Enhanced QTTN Design: Scalable Quantum Circuits for Arbitrary Qubit Counts.
DOI: 10.5220/0013180900003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 3*, pages 766-773
ISBN: 978-989-758-737-5; ISSN: 2184-433X

can efficiently model and optimize the quantum state generated by the ansatz.

There are various tensor network ansatzes available, Matrix Product State (MPS) and Tree Tensor Network (TTN) being among the most widely used. However, current implementations of MPS and TTN are generally applicable only when the number of qubits, n, is in the form $n = 2^x$, where x is an integer (e.g., 1, 2, 3, ...). This requirement limits their applicability, as not all quantum problems require a qubit count that fits this specific pattern. If n does not equal 2^x it becomes necessary to either pad the system with additional qubits or to shrink the network to fit the required qubit count. Padding the system adds unnecessary complexity to the model, introducing extra qubits that do not carry useful information, which can increase the computational resources needed without improving performance. On the other hand, shrinking the network to match the required qubit count often results in a loss of accuracy and effectiveness in the ansatz.

By developing an ansatz that is flexible and scalable without the need for such padding or shrinking, performance can be significantly improved. Such an ansatz would streamline the computational process, making it more efficient and better suited to handling problems with arbitrary qubit counts. Additionally, this flexibility would allow for more efficient utilization of quantum hardware and avoid the pitfalls of excess qubit overhead or reduced expressibility, ultimately enhancing both the accuracy and scalability of the solution.

This work introduces a novel approach for applying tensor-network architectures to problems of arbitrary size, without imposing constraints on the number of qubits. We experimentally evaluate the efficiency of the proposed ansatz through metrics such as fidelity, expressibility, and entanglement strength. Furthermore, we showcase the trainability of our ansatz by implementing a quantum neural network classifier to classify the MNIST handwritten image dataset (Y. LeCun et al., 1998).

1.1 Tree Tensor Network Ansatz

A Tree Tensor Network (TTN) is a hierarchically structured tensor network resembling a tree and can be represented as an acyclic graph T = (G, A) where A denotes tensors (multi-dimensional arrays) connected at their indices, and G represents the graph. An index is a label that connects two or more tensors, representing the shared dimensions across which the tensors interact. In this structure, the nodes correspond to tensors, while the edges represent contracted dimensions between them. Any n-qubit TTN ansatz can be described by n open indices and a tree-like structure.

To construct a TTN that represents a quantum state of dimension 2^n, begin by connecting each of the n open indices to a node, labeling each node uniquely to form the leaves of the tree. At each hierarchical level, group the nodes into pairs, creating two-index tensors to capture local entanglement between the qubits. On ascending the hierarchy, continue merging pairs of tensors at each subsequent level. The new tensors formed will have three indices: two inherited from the previous lower-level tensors and one internal connection to the next level, encapsulating the entanglement between larger groups of qubits. This recursive process is repeated until only a single root tensor remains, which captures the global entanglement of the entire quantum state.

Figure 1 (Guala et al., 2023) illustrates a Tree Tensor Network (TTN) on the left, and its equivalent quantum circuit representation on the right, both corresponding to the process of constructing a quantum ansatz. The nodes $v_0, v_1, v_2, v_3, v_{01}, v_{23}$ represent tensors. v_0, v_1, v_2, v_3 are leaf nodes corresponding to the initial qubits or input tensors. The intermediate nodes v_{01}, v_{23} represent tensors that encapsulate the local correlations between the qubit pairs.The root node v_{13} at the top encapsulates the global entanglement of the entire quantum state, combining all previous layers. The edges between nodes represent tensor contractions, where shared indices are summed over to capture the entanglement between qubits or groups of qubits at different levels of the hierarchy. The circuit on the right reflects the same entanglement structure as the TTN, represented as quantum gates. The input qubits, initialized to $|0\rangle$, correspond to the leaf nodes of the TTN. The first level of gates entangles the initial qubits (matching the first level of tensor nodes in the TTN). Subsequent levels represent further entanglement operations, corresponding to higher levels in the TTN, until the root node is reached.

2 PROPOSED ARBITRARY QUBIT COUNT TTN

We propose a method for constructing quantum circuits using a hierarchical tree structure, designed to handle any number of qubits. At each level of the tree, qubits (or nodes) are paired to establish entanglement.

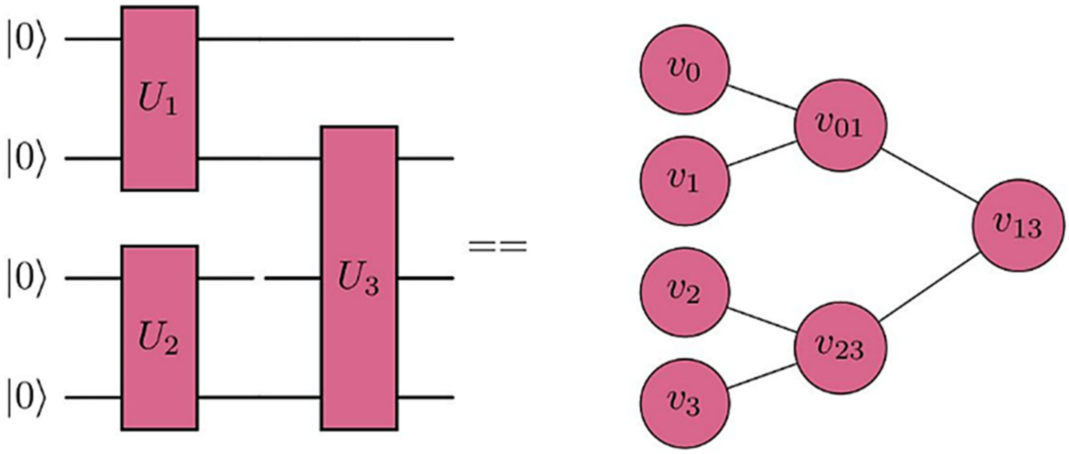

Figure 1: Quantum Circuit (left) and Corresponding TTN (right).

If the number of nodes at any level is odd, that is, one node remains unpaired, we handle this by moving the unpaired node to the next level of the tree.

For instance, if a node labeled v_a does not have a partner at its current level, it is carried over to the next level and renamed v_{aa}, forming a self-paired node. This ensures that every level of the tree has complete pairs, allowing the structure to be progressively built up until the quantum circuit is fully constructed. By systematically pairing nodes and addressing any odd-numbered situations, the method guarantees that qubits are organized into a well-defined entanglement structure, ultimately leading to a fully connected quantum circuit. The process is split into two key modules: (1) **Generate_Tree _Tuples**, which selects the indices for entanglement at each hierarchical level, and (2) **Construct_EQTTN**, which applies unitary operations to form the quantum ansatz.

The **Generate_Tree_Tuples** algorithm takes the number of qubits (*n*) as input and outputs a list of tensor nodes, which defines the connections needed to form the tree tensor network architecture. The algorithm is outlined below.

Algorithm 1

```
Data: number of qubits n
Results: list of list entangled qubits at each level
indices [] []
Initialize tuples_list []
Initialize indices [] []
Assign 0 to i ;
//Generate level 1 /initial entanglement qubit indices
While i is less than n do
    Assign (i, i+1) in tuples_list
    Increment i by 2;
end
 if n is odd
    final node paired with the same node and stored in tuples_list.
end
Store the tuple_list into indices.
// iterate through each level
while len (tuples_list) is equal to 1 do
    Initialize new_tuples [];
    for each i from 0 to len (tuples_list – 1) do
        store (tuples_list[i][1], tuples_list [i + 1][1]) in new_tuples[i]
        increment i by 2;
    end
    if length of tuples_list is odd:
        paired with the same node and store is a final element.
    end
    Store new tuples in tuples_list
    Store indices in new_tuples
end
Return indices
```

Algorithm 1: Finding the entanglement pair list.

Algorithm 1 begins by initializing two lists: tuples_list, which stores node indices for connecting tensors at each level of the network, and indices, which will store all layers of tuples_list. The algorithm first generates pairs of qubit indices by iterating through the range from 0 to num_qubit - 1 in steps of 2, pairing each index with the next; if the number of qubits is odd, the last qubit is paired with itself. These pairs are stored in tuple_list. Then, for each hierarchical layer of entanglement, if tuple_list contains more than one element, the algorithm creates a new_tuple_list by pairing the first indices from consecutive pairs in tuple_list; if there's an odd number of elements, the last element is left unchanged. This new_tuple_list is added to indices, and the process repeats with tuple_list updated to the new pairs. The algorithm stops when only one pair remains in tuple_list and returns indices with all levels of node pairs that define the EQTTN structure.

The Contruct_EQTTN takes the number of qubit and number of layers as input argument and returns a tree tensor network ansatz for arbitrary qubit count. The proposed algorithm is outlined below.

```
Algorithm 2
Data: number of qubits $n$, number of layers $l$.
Results: Quantum circuit qc of size $n$:
Initialize Quantum Circuit   qc of size $n$
Get the entanglement pairs list using Algorithm 1.
Initialize the parameter count by 0
 //Iterate Over Layers
 For each set of layer indices in indices
       For each tuple (i, j) in the layer indices:
             If i is not equal to j:
                Add rotation gates with unique parameters to $i^{th}$ qubit and $j^{th}$ qubit in qc.
                Increment the parameter count by 2
             end
             For each layer from 0 to $l - 1$:
                Add entanglement between $i^{th}$ qubit and $j^{th}$ qubit in qc.
             end
             Add rotation gates with unique parameters to $i^{th}$ qubit and $j^{th}$ qubit in qc.
       end
end
return qc
```

Algorithm 2: Constructing EQTTN ansatz.

The proposed algorithm for constructing the EQTTN ansatz begins by creating a quantum circuit with the specified number of qubits and generating a tree structure using the Generate-Tree-Tuple function to define qubit connections. A parameter counter is initialized to track the number of parameters used. In the first step, parameterized rotation gates (e.g., R_y gates) are applied to each qubit pair at each level of the tree, with the parameter counter incremented by 2 for each distinct pair. In the next step, entanglement gates (such as C_X, C_Z, or C_Y) are added between qubit pairs, followed by additional rotation gates on the entangled qubits, again incrementing the parameter counter by 2. This process continues for all layers of the tree structure. Figure 2 provides an example of a single layer of the 5-qubit EQTTN, built using R_y gates for rotations and C_Z gates for entanglement.

2.1 Parameter Calculations

Let n be the number of qubits and l the number of layers. In an EQTTN, qubits are grouped into pairs at each level of the network. These pairs are then entangled by controlled gates and bound by a pair of parameterized rotation gates, which represent the quantum interactions and correlations between the subsystems. Additionally, there is a layer of parameterized rotation gates applied to each qubit for initialization. Hence the total number of parameters is calculated as follows.

Let t be the total number of pairs per layer which is calculated as follows. Given n qubits, the number of pairs created at level one is $\frac{n}{2}$. This is halved at each subsequent layer until only a single pair remains. The total number of pairs per layer t will be,

$$t = \frac{n}{2} + \frac{n}{4} \dots + 1.$$

t represents the geometric progression (GP) series such that $a, ar, ar^2, ar^3, \dots 1$

$$t = \frac{n}{2} + \frac{n}{2}\left(\frac{1}{2}\right) + \frac{n}{2}\left(\frac{1}{2^2}\right) \dots + 1.$$

Here the first term $a = \frac{n}{2}$ and the common ratio $r = \frac{1}{2}$.

The sum of GP Series for k terms is:

$$S_k = \frac{a(1 - r^k)}{1 - r}.$$

$$t = \frac{\frac{n}{2}\left(1 - (\frac{1}{2})^k\right)}{1 - \frac{1}{2}}.$$

$$p = 2 \times total\ number\ of\ pairs\ per\ layer\ \times \left(\, l + 1(\text{for initialization})\right). \quad (1)$$

$$t = n\left(1 - (\tfrac{1}{2})^k\right)\ . \quad (2)$$

To find k , there are k terms in the series where the k^{th} term is 1

$$ar^{k-1} = 1.$$

Substituting $a, r\ values$

$$\frac{n}{2}\left(\frac{1}{2}^{k-1}\right) = 1.$$

$$\frac{n}{2^k} = 1.$$

$$k = \log\ 2^n.$$

Substitute $k\ in$ (2)

$t = n\ (1 - (\frac{1}{2})^{\log\ 2^{n}})$.

$$t = n\ (1 - (2^{-\log\ 2^{n}})).$$

Substituting $k\ in$ (2)

$$t = n\ (1 - (\tfrac{1}{2})^{\log\ 2^{n}}).$$

$$t = n\ (1 - (2^{-\log\ 2^{n}})).$$

$$t = n\ (1 - (2^{\log\ 2^{\frac{1}{n}}})).$$

$$t = n\ (1 - \frac{1}{n}).$$

$$t = n - 1.$$

Substituting $t\ in$ (1)

$$p = 2\ (n-1)\ (l+1).$$

Thus, the total number of parameters p is calculated.

3 EXPERIMENTS

Variational Quantum Algorithms (VQAs) offer a promising strategy for achieving quantum advantage in the NISQ (Noisy Intermediate-Scale Quantum) era. A key challenge in this approach is designing an efficient ansatz, one that effectively represents the solution space while minimizing circuit depth and the number of parameters to mitigate noise and enhance performance. The quality of an ansatz can be evaluated based on several criteria (Sim et al., 2019): *Expressibility* -the ability of the ansatz to represent a wide range of quantum states, ensuring that the solution space is sufficiently covered for optimization. A highly expressible ansatz can better capture the complexity of the problem being solved. *Entangling Capability*-the ansatz must be able to generate entanglement between qubits, which is essential for exploiting quantum correlations and maximizing quantum computational power. A good ansatz promotes strong entanglement across qubits while maintaining control over circuit depth. *Fidelity* measures the accuracy with which the ansatz can approximate the target quantum state. Higher fidelity indicates that the ansatz is effective in reaching the desired state, despite potential noise and errors in NISQ devices.

Trainability means the ansatz should be easy to optimize using classical optimization methods. Poor trainability can lead to vanishing gradients (barren plateaus), which hinder the learning process. An efficient ansatz should have well-behaved gradients to enable effective optimization. We assess these performance metrics, and the results are shown below.

To compute the **expressibility** of the EQTTN ansatz, we followed a systematic approach to measure how effectively the ansatz explores the quantum state space. First, we generated 1024 samples of pure quantum states from a uniform distribution by applying Haar-random unitary matrices of size $2^{n}\ X\ 2^{n}$, The Haar measure ensures that these random unitary matrices uniformly cover the entire space of possible quantum states, making these samples representative of the full quantum state space. Next, we generated another 1024 samples of quantum states using our EQTTN ansatz with various random parameter settings. Then we calculated their differences and obtained a distribution A. The expressibility of the ansatz was then quantified by calculating the norm of A, providing a measure of how closely the ansatz-generated states resemble the uniformly distributed Haar-random states. The resulting expressibility measure ranges from 0 to 1, where a value near 0 indicates high expressibility (i.e., the ansatz can explore the state space effectively and uniformly), while a value near 1 suggests lower expressibility, meaning the ansatz is more constrained in the states it can generate.

We performed this analysis across several ansatz

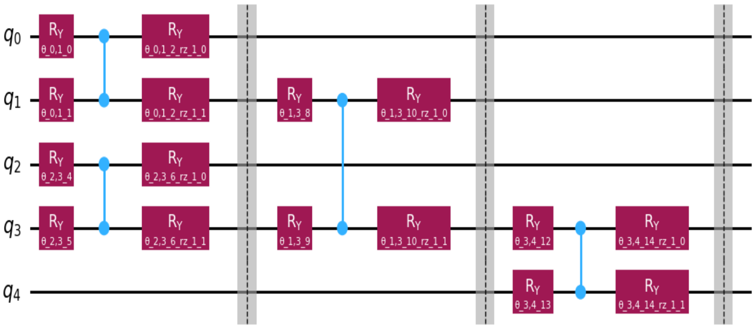

Figure 2: Single layer EQTTN for 5-qubits.

types available in the Qiskit library (IBM), including Two_local, N_local, EfficientSU2, TTN which are standard quantum circuit ansatz types used in variational quantum algorithms, each designed to balance flexibility and computational complexity. These approaches range from simple operations (Two_local, N_local which are parameterized circuits consisting of alternating rotation layers and entanglement layers) to more structured, hierarchical designs (Tree Tensor Network or TTN) and efficient representations of unitary transformations which is our proposed EQTTN ansatz. As depicted in Figure 3, our EQTTN ansatz demonstrates exceptional expressibility compared to the others. Furthermore, Figure 4 provides a visual representation of the expressibility of a 5-qubit single-layer EQTTN ansatz plotted on the Bloch sphere, which illustrates its extensive coverage of the quantum state space.

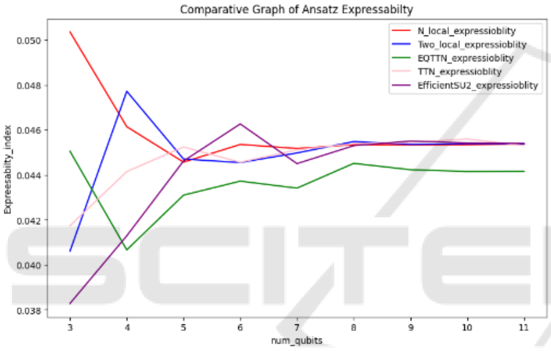

Figure 3: Comparison graph for expressiblity.

Entangling Capability: Quantum correlations, particularly entanglement, play a key role in the effectiveness of Variational Quantum Algorithms (VQAs). In the Variational Quantum Eigen solver (VQE), entanglement enables accurate representation of electron correlations essential for finding ground state energies in quantum chemistry (Hubregtsen et al., 2020). In quantum artificial intelligence, entanglement enhances the learning capacity of models by representing complex data correlations Bengtsson et al.,2017, Huggins et al.,2019).

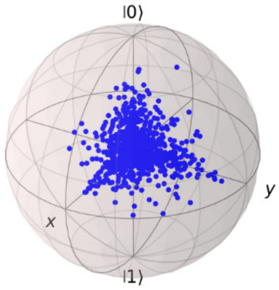

Figure 4: 5-qubit single layer ETTN expressiablity.

We quantify the entangling capability by using the Meyer-Wallach measure (Meyer et al., 2002), a scalable entanglement measure for multi-particle (multi-qubit) systems. We took the sample of 1024 quantum states with random parameters from a 5-qubit single layer ETTN and then found the Meyer-Wallach measure of each quantum state and then found the average which gives the entanglement capability of our proposed ansatz. This measure ranges from 0 to 1. A value close to 1 indicates that the ansatz has higher entangling capabilities. Our 5 qubit EQTTN Gives at an average of 60%. Figure 5 shows the comparative graph for entanglement.

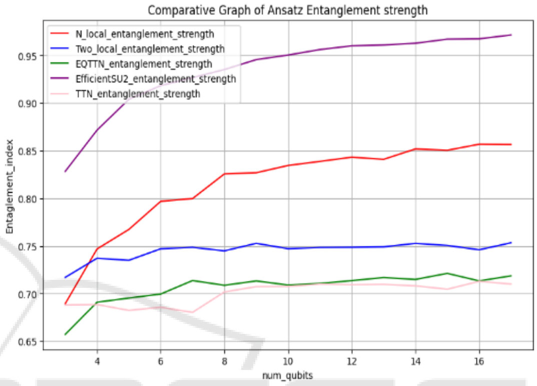

Figure 5: Comparison graph for entanglement strength.

Fidelity is a widely recognized metric in quantum computing for assessing the accuracy of quantum operations, particularly in the presence of noise. It shows how close the actual quantum state is to the ideal, error-free state. Fidelity measures the similarity between the ideal density matrix (representing the expected quantum state) and the actual density matrix obtained after executing a quantum operation. A high-fidelity score indicates that the actual state closely approximates the ideal state, providing an indicator of the performance and reliability of quantum gates and circuits.

In our study, we conducted fidelity experiments using various ansatz types, including Two-Local, N-Local, EfficientSU2, TTN and our proposed EQTTN, ansatz. For a 5-qubit single-layer architecture, as shown in Table 1, the EQTTN ansatz consistently outperformed the others, achieving the highest fidelity. This superior fidelity demonstrates the robustness and accuracy of the EQTTN ansatz in preserving the desired quantum states, making it a highly promising candidate for noise-resilient quantum operations.

Table 1: Fidelity results.

No	Ansatz	Fidelity
1	N_local	0.52739
2.	EQTTN	0.79013
3.	Two_local	0.65312
4	EfficientSU2	0.75756
5.	TTN	0.4361

Trainability: To demonstrate the trainability of the proposed ansatz, we designed a Quantum Neural Network (QNN) classifier to perform digit classification on the MNIST dataset (Y. LeCun et al., 1998). We used a 5-qubit single-layer EQTTN (Enhanced Quantum Tree Tensor Network) ansatz, implemented using Qiskit's State Vector Simulator (IBM)For encoding the classical MNIST data into the quantum states, we employed amplitude encoding, which efficiently maps the input data into the amplitudes of a quantum state. This encoding method allows us to handle high-dimensional data using fewer qubits.

We used 10000 samples from the MNIST data set as the training set and 2000 samples for testing. The model's loss function was defined as the cross-entropy loss, which is a standard choice for classification problems as it measures the dissimilarity between the predicted and actual labels. The model parameters were optimized using a gradient based optimizer (such as COBIYALA), which updates the parameters to minimize the loss. Table 2 illustrates the results of accuracy and time to converge the loss function for our 5 different ansatzes

Based on the results we have shown here, our EQTTN is the best ansatzes among the various ansatz, balancing performance across key metrics. While its entanglement strength is moderate compared to N_local and EfficientSU2, EQTTN exhibits stability and consistency across varying qubit numbers, which is crucial for robust quantum computations. In terms of expressibility, EQTTN shows a smooth and stable curve, outperforming other ansatz in consistency as qubits increase, making it reliable for generalization across different problem sizes. Its fidelity score of 0.79013 is the highest, reflecting its strong capability in accurately representing quantum states. Moreover, EQTTN achieves competitive trainability with an average training accuracy of 96.22% and testing accuracy of 96.83%, coupled with a reasonable convergence time of 1172.64 seconds, making it more efficient than EfficientSU2 but slightly slower than Two_local and N_local. Overall, EQTTN offers a solid balance of accuracy, expressibility, and computational efficiency, making it a well-rounded choice for quantum machine learning applications.

4 CONCLUSIONS

In this paper, we present an Enhanced Quantum Tree Tensor Network (EQTTN) ansatz designed for Variational Quantum Algorithms (VQAs) to scale efficiently with arbitrary qubit counts.

The performance of the EQTTN ansatz is evaluated based on several key metrics: expressability, entanglement strength, fidelity, and trainability. Our results demonstrate that the EQTTN ansatz offers superior performance. The proposed approach leads to better computational performance as it calculates tensor network anstazes for n cubits as opposed to 2^n qubits. If n is large, the improvement in performance will be significant. In future work, we aim to explore the integration of the EQTTN ansatz with error-mitigation techniques to further enhance. its robustness in noisy quantum environments. Additionally, extending the framework to investigating its application in real-world problems could provide valuable insights.

Table 2: Trainability results.

No	Ansatz	Average_acuuracy in percentile		Average_conv.time in secs
		Training_set	Testing_set	
1	N_local	94.128	94.42	899.32
2.	EQTTN	96.22	96.83	1172.64
3.	Two_local	97.34	97.78	634.81
4	EfficientSU2	96.121	96.456	2529.31
5.	TTN	97.38	97.42	4129.24

REFERENCES

Cerezo, M., Arrasmith, A., Babbush, R. et al. Variational quantum algorithms. Nat Rev Phys 3, 625–644 (2021). https://doi.org/10.1038/s42254-021-00348-9

Guala, D., Zhang, S., Cruz, E. et al. Practical overview of image classification with tensor-network quantum circuits. Sci Rep 13, 4427 (2023). https://doi.org/10.1038/s41598-023-30258-y

Sim, Sukin, Peter D. Johnson and Alán Aspuru-Guzik. (2019) Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. Advanced Quantum Technologies 2019

Bengtsson, I., & Życzkowski, K. (2017). Geometry of Quantum States: An Introduction to Quantum Entanglement. Cambridge University Press.

Huggins, William & Patil, Piyush & Mitchell, Bradley & Whaley, K & Stoudenmire, Edwin. (2019). Towards Quantum Machine Learning with Tensor Networks. Quantum Science and Technology. 4. 10.1088/2058-9565/aaea94.

Nielsen, M. A., & Chuang, I. L. (2011). Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press.

Meyer, David A., and Nolan R. Wallach. (2002). Global Entanglement in Multiparticle Systems. Journal of Mathematical Physics 43 (9):4273–78. https://doi.org/10.1063/1.1497700

Hubregtsen, T., Pichlmeier, J. and Bertels, K., 2020. Evaluation of Parameterized Quantum Circuits: on the design, and the relation between classification accuracy, expressibility and entangling capability. arXiv preprint arXiv:2003.09887.

H.-M. Rieser, F. Köster, and A. P. Raulf, (2023), Tensor networks for quantum machine learning, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 479, no. 2275, Jul. 2023.

Seitz P, Medina I, Cruz E, Huang Q, Mendl CB. Simulating quantum circuits using tree tensor networks. Quantum. 2023 Mar 30; 7:964.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998. IBMquantumexperiencehttps://quantum.ibm.com/ Simulation tool.