



Application of Quantum Tensor Networks for Protein Classification

Debarshi Kundu

Pennsylvania State University
State College, PA, USA
dqk5620@psu.edu

Jian Wang

Pennsylvania State University
Hershey, PA, USA
juw1179@psu.edu

Archisman Ghosh

Pennsylvania State University
State College, PA, USA
apg6127@psu.edu

Nikolay Dokholyan

Pennsylvania State University
Hershey, PA, USA
nxd338@psu.edu

Srinivasan Ekambararam

Pennsylvania State University
Hershey, PA, USA
sje5459@psu.edu

Swaroop Ghosh

Pennsylvania State University
State College, PA, USA
szg212@psu.edu

ABSTRACT

Computational methods in drug discovery significantly reduce both time and experimental costs. Nonetheless, certain computational tasks in drug discovery can be daunting with classical computing techniques which can be potentially overcome using quantum computing. A crucial task within this domain involves the functional classification of proteins. However, a challenge lies in adequately representing lengthy protein sequences given the limited number of qubits available in existing noisy quantum computers. We show that protein sequences can be thought of as sentences in natural language processing and can be parsed using the existing Quantum Natural Language framework into parameterized quantum circuits of reasonable qubits, which can be trained to solve various protein-related machine-learning problems. We classify proteins based on their sub-cellular locations—a pivotal task in bioinformatics that is key to understanding biological processes and disease mechanisms. Leveraging the quantum-enhanced processing capabilities, we demonstrate that Quantum Tensor Networks (QTN) can effectively handle the complexity and diversity of protein sequences. We present a detailed methodology that adapts QTN architectures to the nuanced requirements of protein data, supported by comprehensive experimental results. We demonstrate two distinct QTNs, inspired by classical recurrent neural networks (RNN) and convolutional neural networks (CNN), to solve the binary classification task mentioned above. Our top-performing quantum model has achieved a 94% accuracy rate, which is comparable to the performance of a classical model that uses the ESM2 protein language model embeddings. It's noteworthy that the ESM2 model is extremely large, containing 8 million parameters in its smallest configuration, whereas our best quantum model requires only around 800 parameters. We demonstrate that these hybrid models exhibit promising performance, showcasing their potential to compete with classical models of similar complexity.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0605-9/24/06
<https://doi.org/10.1145/3649476.3658701>

KEYWORDS

Quantum Machine Learning, Quantum Neural Network, Protein Sequence Classification

ACM Reference Format:

Debarshi Kundu, Archisman Ghosh, Srinivasan Ekambararam, Jian Wang, Nikolay Dokholyan, and Swaroop Ghosh. 2024. Application of Quantum Tensor Networks for Protein Classification. In *Great Lakes Symposium on VLSI 2024 (GLSVLSI '24)*, June 12–14, 2024, Clearwater, FL, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3649476.3658701>

1 INTRODUCTION

Machine learning for protein engineering: The use of computational methods, particularly machine learning (ML), has dramatically accelerated drug discovery timelines, especially in protein design. Innovations like ESM-2[11] and AlphaFold[4] have transformed protein sequence analysis and structure prediction, respectively, with AlphaFold achieving near-experimental accuracy as demonstrated in the CASP competitions. These advancements highlight the power of ML to deepen our understanding of protein structures and functions by utilizing large datasets to uncover new predictive relationships. However, the complexity of certain computational biology challenges still surpasses the reach of classical computing, pointing towards the potential of quantum machine learning to address these issues more effectively.

Quantum Natural Language Processing (QNLP): Quantum computing applied to Natural Language Processing (NLP) uses word embeddings as quantum states for faster classification. Word embeddings in quantum NLP are formed through linear mappings into tensor products. The main trainable part of quantum NLP circuits is Parameterized Quantum Circuits (PQCs), featuring entangling multi-qubit operations and single-qubit rotations to explore solution spaces. These circuits function as Quantum Tensor Networks, breaking down sentences into networks of simpler tensors for efficient classification.

Protein Sequence as a sentence: Viewing a protein sequence as a sentence, with each amino acid acting as a word, presents a powerful analogy for deciphering the structure and function of proteins [14]. This perspective highlights the complexity and specificity inherent in protein sequences, likening a protein to a carefully composed sentence where the amino acids are arranged in a precise sequence. Each "word" (amino acid) adds its unique characteristics to the "sentence" (protein sequence), influencing the

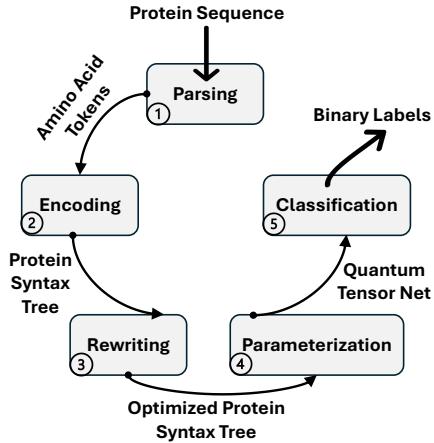


Figure 1: A diagram describing the flow of binary classification of the protein sequence.

protein's folding, structure, and function within biological systems. For example, the positioning of amino acids such as lysine, arginine, and glutamate within a protein can be seen as words constructing a sentence, where the exact sequence and arrangement are critical for the protein's capability to fulfill its designated roles. Similarly, altering a word in a sentence can change its entire meaning, just as modifying an amino acid in a protein sequence can drastically affect the protein's functionality. This analogy emphasizes the critical nature of sequence fidelity in proteins, underscoring the finely tuned equilibrium of biological systems where each "word" plays a pivotal role in the "narrative" of life [20].

Motivation: Integrating Quantum Natural Language Processing (QNLP) into protein sequence analysis offers the potential to revolutionize drug discovery and our understanding of biological processes. This approach leverages quantum computing's capacity for high-dimensional space management, speed, and semantic analysis, enabling more accurate predictions of protein functions, structures, and interactions. By viewing amino acid sequences as sentences, Quantum NLP can provide deeper insights into the protein "language," surpassing current bioinformatic tools in sequence alignment, functional motif identification, and protein function annotation[9], [7]. This integration could significantly advance drug discovery by exploiting quantum algorithms for enhanced efficiency and speed. However, there is a need to represent the long protein sequences in the quantum circuits with reasonably small qubits and reasonably deep quantum circuits in the NISQ-era quantum computers. There is a need to extract the signal from these sequences to solve important challenges in drug discovery.

Contributions: Building on the foundational work by [6], our study marks the proof of concept of classifying long protein sequences leveraging Quantum Tensor Networks (QTNs). This advancement is pivotal, as it not only extends the applicability of QTNs beyond their traditional domains but also introduces a novel methodology for handling the complexities inherent in protein sequence data. The utilization of quantum computing in this context is not merely for its computational prowess but also for its ability to capture the intricate patterns and relationships within

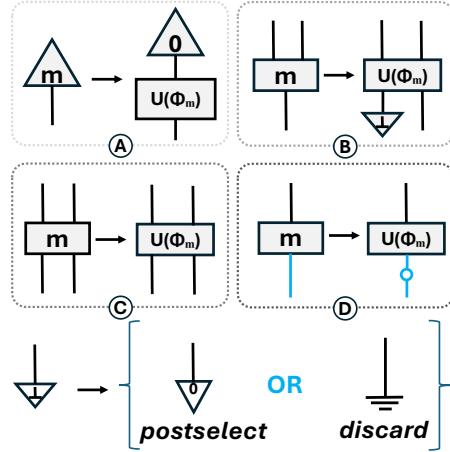


Figure 2: Assignment of parameterized quantum circuits $U(\phi_i)$ to boxes labeled i : This is an example of a function definition, where the words are mapped to qubits (bits in case of blue wires) via the unitary matrices $U(\phi)$. The \perp is either represented as an all-zeroes state, postselect, or discard.

biological sequences, which are often beyond the reach of classical computational techniques.

In a nutshell, our contributions to this paper are as follows: (a) we have successfully demonstrated, for the first time, the potential of QTNs[6] in the classification of long protein sequences, (b) we showed that QTNs, inspired by convolutional and recurrent neural networks, are capable of learning representations of proteins using a relatively small qubit circuit and (c) our findings underscore a significant advancement over classical models, showcasing the inherent advantages of quantum computing in processing and classifying biological data. This comparison not only validates the effectiveness of our approach but also sets the stage for future explorations into quantum bioinformatics.

2 PROPOSED MODELS

Here we first explain the entire process of generating a parametrized quantum circuit from a protein sequence and training the circuits for the binary classification task (Fig. 1). Then we explain two important steps to develop the quantum models namely, (a) building compositional schemes to convert protein sequences into networks which are represented using wires and boxes and (b) defining semantic functor to map these networks into quantum circuits.

Protein to quantum model pipeline: The process begins with a protein sequence input that undergoes parsing using a state-of-the-art neural-trained parser, resulting in a protein syntax tree. This tree is encoded into a string diagram, abstractly representing the relationships between elements in the sequence. These string diagrams are based on category theory [5] and can be simplified by rewriting rules to reduce redundancy and adapt the computation for quantum processors. After rewriting, the diagrams are parameterized and converted into a quantum circuit using specific parameterization schemes and ansätze choices. The quantum circuit is then classified using a QTN, resulting in binary labels, and is ready for training.

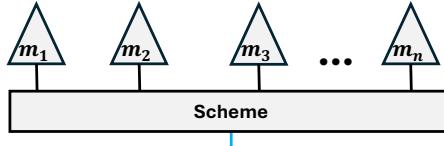


Figure 3: A sentence is broken into words $m_1 \dots m_n$, converted to corresponding unitaries based on the Functor rules in Fig. 2 and finally run through the Scheme (QTN).

This entire model is structured based on the grammatical construction of the input sequence and is optimized for implementation on quantum processing units.

Compositional Schemes: A compositional scheme is initially defined for a given sequence, employing the graphical language of process theories. The processes within these schemes are represented by boxes, which have input and output wires. These wires carry types - either the 'internal' type τ or the 'sentence' type σ . The composition of these boxes, following type constraints, allows for the generation of process diagrams that represent the scheme for sequence analysis. Given a vocabulary $\mathcal{V} = \{m_i\}_i$ comprising a finite set of words (or tokens), we consider compositional schemes for sequences \mathcal{S} of finite length over this vocabulary. The schemes are then semantically mapped onto QTN models (Fig. 2).

Semantic Functor \mathcal{F} : It is a structure-preserving map that assigns Hilbert space semantics to the compositional schemes, thereby designing parameterized quantum circuits (PQCs) for various components of the scheme. For handling non-deterministic outcomes, two strategies are defined: *postselect* and *discard*. *Postselect* involves conditioning on a particular measurement outcome, typically the all-zeros state, while *discard* involves ignoring specific dimensions of the quantum state like a partial trace.

Parameterized Quantum Circuits (PQCs) for QTNs: The PQCs designed by \mathcal{F} involves **Word-State Preparation** (Fig. 2A): These boxes prepare a word-state of type $\tau^{\otimes 0} \rightarrow \tau^{\otimes 1}$, which is associated with a parameterized quantum state prepared by applying the circuit $U(\phi_m)$ to the fixed input state $|0\rangle^{\otimes q}$. Each word corresponds to a unique set of parameters ϕ_m , **Filter Application** (Fig. 2B): The filter boxes, with type $\tau^{\otimes 2} \rightarrow \tau^{\otimes 2}$, are associated with a $U(\phi_m)$ operating on 2^q input and 2^q output qubits, simulating the filtering process within the protein sequence, **Merge Operation** (Fig. 2C): The m-box, typed $\tau^{\otimes 2} \rightarrow \tau^{\otimes 1}$, is mapped to $U(\phi_m)$, with 2^q input qubits and q output qubits. The reduction in qubits is achieved by either discarding or postselecting the redundant qubits via the \perp -effect, and **Classification** (Fig. 2D): The classifier boxes, typed $\tau^{\otimes 1} \rightarrow \sigma^{\otimes 1}$, is associated with a unitary $U(\phi_m)$, which processes a q -qubit state input and outputs a q' -qubit state. This state is subsequently measured in the Z basis, yielding a vector in $[0, 1]^{2^{q'}}$, representing the classification outcome based on the Born rule probabilities. These strategies influence the tensor network topology, allowing for efficient tensor contraction which encapsulates the sequence processing task within a quantum framework.

Example: A protein sequence is represented as a sentence and broken down into independent *words* and then run through the QTN (Fig. 3). We dive deep into the idea by considering a protein sequence of length four (*AGSQ*) in Fig. 4 and define the *aminoacids*

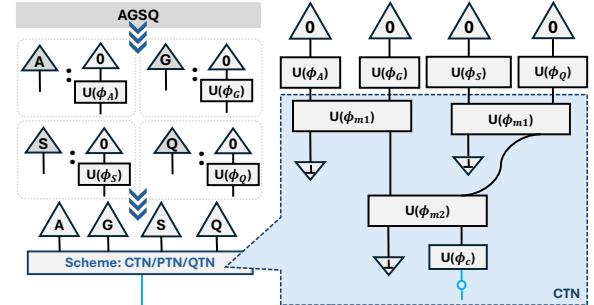


Figure 4: An example to parse an demonstrative protein sequence *AGSQ* into a protein syntax tree based on CTN.

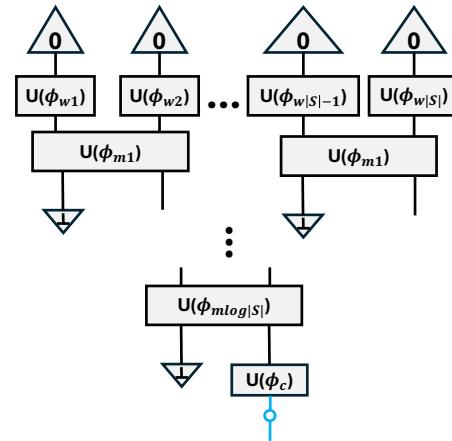


Figure 5: Convolutional tensor network (CTN)

as different *words* that are then converted to the respective unitaries (parameterized input states) based on the rules of the Functor mapping (Fig. 2). These mapped states finally culminate into the Quantum Convolution Tensor Net (Fig. 5).

2.1 Compositional scheme: Recurrent Neural Net inspired

We examine QTNs with a model that adheres to the sequential flow akin to the natural progression of words in a text. This model, derived by implementing the semantic functor \mathcal{F} outlined in Fig. 2, results in what we refer to as the path tensor network model, symbolized as $\mathcal{F}(\phi)[\text{path}] = \text{PTN}$, and illustrated in Fig. 6. This approach sequentially aligns with the reading order, mapping out a straightforward path through the sequence.

2.2 Convolutional Tensor Networks

Hierarchical and Uniform PTN Models: Delving deeper, we introduce a nuanced layer to the model by attributing a hierarchical structure to the parameter sets $\{\phi_{mi}\}$, contingent upon their sequential position i , within the range of $\{1, 2, \dots, |S| - 1\}$. This adjustment births the hierarchical PTN (hPTN) models. Proceeding further, we harmonize the parameters across all merging circuits (m -circuits) to a singular set, $\phi_m = \phi_{mi}$ for all i , thereby engendering a recurrent structure, which we term the uniform PTN (uPTN). This uPTN model distills down to a basic form of a recurrent quantum model, or equivalently, a matrix product state (MPS) model, in

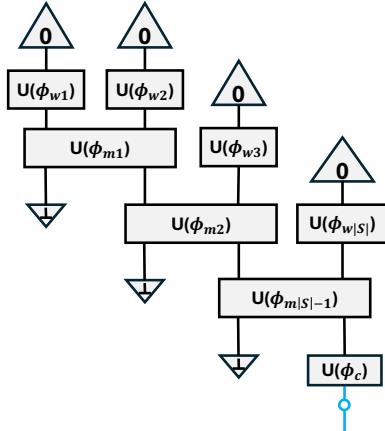


Figure 6: Path tensor network (PTN)

which all dimensions except for the last are either disregarded or selected based on outcomes, with the remaining dimension capturing the overall semantic value of the sentence. This initial compositional approach intentionally bypasses considerations of syntax and distant correlations, positioning it as an elementary framework for juxtaposition with models that incorporate syntactic awareness. While simplistic, this model is built upon a principle of local compositional application, serving as a critical benchmark.

Our approach uses a refined compositional scheme, similar to convolutional neural networks, which serves as an advancement over the basic tree structure. This scheme is ingeniously crafted by integrating additional layers of filtering boxes (f-boxes) into the tree architecture. These f-boxes are strategically placed to operate prior to the merging boxes (m-boxes) along adjacent wires that do not converge into the same m-box. This setup results in a convolutional tensor network (CTN), distinguished by its ability to filter out superfluous entanglements at each layer through the f-circuits, followed by a consolidation of qubit wires by the m-circuits. This process effectively distills the sequence, preserving only the essential information pertinent to the designated task. **Hierarchical and Uniform Variants:** The CTN model evolves into hierarchical (hCTN) and uniform (uCTN) variants based on the distribution and uniformity of the parameter sets across the layers. The hierarchical model shares parameter sets within the same layer, whereas the uniform model extends this sharing across the entire model.

2.3 Classical Model

To establish a baseline for rigorously assessing the capabilities of our quantum models, we developed a classical model architecture (Fig. 7) that exploits deep learning techniques, specifically tailored for processing and classifying protein sequences. Central to our classical model is the integration of embeddings derived from the ESM2 [11] pretrained model. Esteemed as a cutting-edge development in machine learning for bioinformatics, ESM2 is intricately designed to distill meaningful features from protein sequences, thereby representing a substantial leap forward in our ability to capture the complex patterns and functional attributes inherent in proteins. This model's ability to learn from an extensive compendium of protein sequences endows it with the capacity to abstract a profound

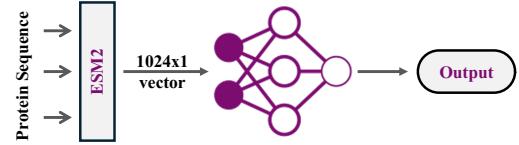


Figure 7: Protein sequence is fed into the ESM2 embedding that passes through a fully connected classical layer to classify the input sequence.

representation of amino acid interrelations, structural motifs, and other critical biochemical properties.

Within our classical framework, each protein sequence undergoes initial processing by the ESM2 model, yielding a fixed-size (1024) embedding vector. This vector serves as a condensed representation of the sequence's biological and contextual nuances, primed for subsequent analysis via neural network techniques. These embeddings are then channeled into a fully connected network, comprising three hidden layers of size (512, 256, 128) of interconnected neurons. The training regimen for this model utilizes the same labeled dataset of protein sequences mentioned earlier, employing binary cross-entropy as the loss function to fine-tune the network's weights. Adam, renowned for its optimization efficacy [10], was the algorithm of choice for this process, facilitating efficient and effective model training. By juxtaposing this classical model's performance against that of our quantum approaches, we endeavor to elucidate the quantum computing paradigm's potential benefits and efficacy in tackling the challenges associated with protein sequence classification.

3 METHODOLOGY

3.1 Dataset

Compilation: The dataset has been compiled using protein sequences obtained from UniProt [1]. The dataset of human protein sequences has been cleaned and preprocessed to have 80 to 200 amino acids in each sequence. The protein sequences have a categorization that has been based on subcellular localization into two groups—proteins in the cytosol or cytoplasm and those associated with the cell membrane. **Structure:** The dataset is made of 1136 protein sequences after preprocessing which is divided into training, validation, and testing data subsets having 980, 123, and 123 protein sequences respectively. The size of the dataset has been scaled down to cater to the long runtime of quantum simulations while ensuring model validation and a thorough evaluation of its predictive performance. **Significance:** The dataset comprises protein sequences that can be classified based on their location in the cell (cytoplasm or cell membrane), therefore facilitating a deeper understanding of the functional implications of proteins based on their cellular locales. On a broader scale, it contributes to the study of complex cellular functions, and biological processes.

3.2 Implementation Details

The Lambeq [8] library, a forefront tool in quantum natural language processing (QNLP), introduces a sophisticated method for translating diagrammatic representations of linguistic structures into quantum circuits, enabling the exploration of various quantum

ansatzes for processing and analysis. The ansatzes explored include the IQPAnsatz, Sim14Ansatz, Sim15Ansatz, and MPSAnsatz, each offering unique advantages for different types of quantum computations [18] as explained next. **IQPAnsatz (Instantaneous Quantum Polynomial-time Ansatz):** This ansatz constructs circuits that are believed to implement computations not efficiently simulatable by classical computers, focusing on problems that can be encoded in a certain polynomial structure. It is particularly suited for tasks where quantum advantage is explored.

The mapping from diagrammatic representations to quantum circuits within `lambeq` is facilitated by parsers like `spiders_reader`, `cups_reader`, and `stairs_reader` [8]. `spiders_reader` translates complex syntactic interactions into quantum circuits using graphical elements known as spiders, facilitating the representation of non-linear word relationships. `cups_reader` captures entanglement between elements in a sentence through cups in diagrammatic notation, effectively modeling pairwise dependencies. `stairs_reader` leverages a staircase pattern to represent dependencies of elements within a sentence, ideal for capturing long-range contextual information crucial in understanding protein sequences.

The transition from these diagrammatic representations to quantum circuits involves encoding linguistic or biological data as initial quantum states, followed by the application of quantum gates as dictated by the chosen ansatz. This process effectively translates the structure and semantics of the input data into a form amenable to quantum computation, enabling the exploration of quantum mechanical advantages in processing complex sequences. Simulation and training of these models are facilitated through the `tensofnetwork` library and JAX, respectively, with the latter enabling Just-In-Time compilation for efficient processing [16] [3]. Among various quantum ansatz (IQPAnsatz, Sim14Ansatz, Sim15Ansatz [18], MPSAnsatz), the expressive ansatz 14 was selected for its notable test performance [18]. Optimization is achieved using AdamW, with the aim of minimizing binary cross-entropy loss for accurate label prediction [12]. Considering the prospect of quantum computer training, we suggest the parameter-shift rule for gradient estimation or the use of SPSA for its practicality in near-term quantum computing environments [2, 17, 19].

The model selection process uses k-fold validation and early stopping, focusing on hyperparameters like embedding qubit count, ansatz depth (q, D), and learning rate. In k-fold validation, the data is split into k parts, training on k-1 and validating on the remaining part, iteratively. This ensures a comprehensive evaluation. The model's generalization is finally tested on an unseen dataset, assessing prediction accuracy. We ensure a consistent comparison framework by reporting test accuracies at the peak of validation accuracy for specified hyperparameters and learning rate settings, under a fixed seed for reproducibility.

4 RESULTS

Our model follows an encoder architecture, which essentially means they accept sentences as inputs and generate corresponding outputs, thereby serving the role of classifiers. These models are predominantly aimed at binary classification tasks, involving the measurement of a singular qubit ($q' = 1$) from the output quantum state produced by the U_c circuit. This process determines probabilities for two possible outcomes, p_0 and p_1 by measuring the average

Table 1: Test accuracy for evaluated models under different measurement procedures (discard) and (postselect)

	uPTN	hPTN	uCTN	hCTN
discard	0.73	0.94	0.57	0.71
postselect	0.72	0.83	0.60	0.67

Table 2: F1-score for evaluated models under different measurement procedures (discard) and (postselect)

	uPTN	hPTN	uCTN	hCTN
discard	0.86	0.97	0.71	0.82
postselect	0.83	0.84	0.74	0.80

state of the qubits, with each outcome directly mapping to a class label. The tree-like design of our introduced model species promotes not only efficient computation but also a natural resistance to the occurrence of barren plateaus during the training process, a notable hurdle in the optimization of quantum models [13, 15, 21].

We have explored different Quantum Tensor Networks, viz., the Path Tensor Network (PTN) and the Convolutional Tensor Network (CTN)—with hierarchical and uniform parameter-sharing strategies. We analyze the performance of four models: the uniform path tensor network (uPTN), the hierarchical path tensor network (hPTN), the uniform convolutional tensor network (uCTN), and the hierarchical convolutional tensor network (hCTN).

Our results, summarized in two tables (Table 1 and 2), provide the test accuracy and F1-scores for each model, with and without post-selection. The hPTN model outperforms the others in both metrics significantly, indicating its superior capability in capturing the necessary features for classification. The uPTN follows, showing decent performance, but not matching the hierarchical counterpart. The uCTN and hCTN models exhibit lower performance, with uCTN scoring the lowest on both accuracy and F1-score.

Model Performance Analysis: The PTN-based models have a sequential architecture that aligns well with the natural sequence of amino acids in protein sequences allowing better classification accuracy. In **hPTN** models (Fig. 8), the sequential information of protein sequences along with the varying dependencies are captured very effectively due to the presence of unique parameter sets in the model. **uPTN** models, based on recurrent quantum nets, perform consistently well but are limited in flexibility by a single shared parameter set across all PQCs. Although **CTNs** have enhanced tree-like architecture, they are not very adept at capturing the long-range correlations effectively. The slight improvement of the performance of **hCTN** models over **uCTN** models suggests that having a hierarchical unique parameter set helps in classification more than having a single shared parameter set. In the comparative analysis of quantum models for protein sequence classification, the ESM2-based classical model (Fig. 7), with its high accuracy of 0.98, sets a significant benchmark for performance. Notably, the hierarchical Path Tensor Network (hPTN) quantum model closely rivals this benchmark with an impressive accuracy of 0.94. This near-parity highlights the substantial potential of quantum models to reach and possibly exceed the performance standards set by advanced classical models in complex biological computations.

Model runtime analysis: CTN models, characterized by their substantial parameter count, require significantly longer training durations, approximately 8 hours per epoch. Conversely, PTN models exhibit a markedly swifter training pace, completing the entire

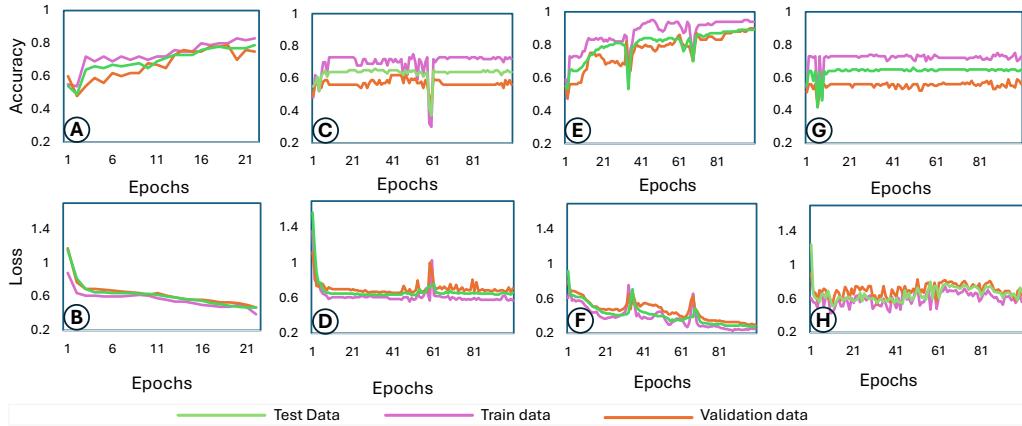


Figure 8: Accuracy and Loss plots for training, test, and validation data evaluated for PTN models under postselect and discard measurements. Plots A-D represent postselect measurement for hierarchical (A, B) and uniform (C, D) PTNs. Plots E-H represent discard measurement for hierarchical (E, F) and uniform (G, H) PTNs.

simulation in about 1 hour. Intriguingly, CTN models achieve convergence in fewer epochs (around 4 to 5) compared to PTN which took much higher number of epochs as shown in Fig. 8. Despite the classical model being trained within an hour, it's important to acknowledge that the foundational pre-trained protein language model, ESM2, underwent a training process powered by extensive computational resources for nearly a week.

Limitations: The QTNs were evaluated under idealized conditions devoid of quantum noise, likely leading to an optimistic representation of their capabilities. Such an omission of quantum noise considerations prevents a fully equitable comparison to the classical ESM2 model and overlooks the challenges posed by the hardware noise in the devices of the current noisy quantum computers. Future research should prioritize addressing this gap by incorporating noise models in the simulation backends.

5 CONCLUSION

We proposed and evaluated four flavors of Quantum Tensor Nets (QTNs) namely, hierarchical Path Tensor Network (hPTN), uniform Path Tensor Network (uPTN), hierarchical Convolutional Tensor Network (hCTN), and uniform Convolutional Tensor Network (uCTN) to classify protein sequences based on their cellular locales (cytosol/cytoplasm or cell membrane).

The hPTN model demonstrated superior performance in classifying protein sequences compared to its uniform counterpart and the convolutional tensor network variants. The uniform models, particularly the uCTN, require further investigation and possible architectural or parameter adjustments to improve their performance.

ACKNOWLEDGMENTS

We extend our gratitude to Avimita Chatterjee (Pennsylvania State University) for her valuable insights and assistance in structuring the work. The work is supported in parts by the National Science Foundation (NSF) (CNS-1722557, CCF-1718474, OIA-2040667, DGE-1723687, and DGE-1821766) and Intel's gift.

REFERENCES

- [1] 2023. UniProt: the universal protein knowledgebase in 2023. *Nucleic Acids Research* 51, D1 (2023), D523–D531.
- [2] Xavier Bonet-Monroig et al. 2023. Performance comparison of optimization methods on variational quantum algorithms. *Physical Review A* 107, 3 (Mar 2023). <https://doi.org/10.1103/physreva.107.032407>
- [3] James Bradbury et al. 2018. *JAX: composable transformations of Python+NumPy programs*. <http://github.com/google/jax>
- [4] Patrick Bryant et al. 2022. Improved prediction of protein-protein interactions using AlphaFold2. *Nature communications* 13, 1 (2022), 1265.
- [5] Wojciech Buszkowski. 2001. Lambek Grammars Based on Pregroups. In *Logical Aspects of Computational Linguistics*, Philippe de Groote, Glyn Morrill, and Christian Retoré (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 95–109.
- [6] Carys Harvey et al. 2023. Sequence Processing with Quantum Tensor Networks. *arXiv preprint arXiv:2308.07865* (2023).
- [7] Lloyd CL Hollenberg. 2000. Fast quantum search algorithms in protein sequence comparisons: Quantum bioinformatics. *Physical Review E* 62, 5 (2000), 7532.
- [8] Dimitri Kartsaklis et al. 2021. lambeq: An Efficient High-Level Python Library for Quantum NLP. *arXiv preprint arXiv:2110.04236* (2021).
- [9] Mohammad Hassan Khatami et al. 2023. Gate-based quantum computing for protein design. *PLOS Computational Biology* 19, 4 (2023), e1011033.
- [10] Diederik P Kingma et al. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [11] Zeming Lin et al. 2023. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 379, 6637 (2023), 1123–1130.
- [12] Ilya Loshchilov et al. 2017. Decoupled Weight Decay Regularization. *arXiv:1711.05101* [cs.LG]
- [13] Cervero Martín et al. 2023. Barren plateaus in quantum tensor network optimization. *Quantum* 7 (Apr 2023), 974. <https://doi.org/10.22331/q-2023-04-13-974>
- [14] Ofer et al. 2021. The language of proteins: NLP, machine learning & protein sequences. *Computational and Structural Biotechnology Journal* 19 (2021), 1750–1758.
- [15] Arthur Pesah et al. 2021. Absence of barren plateaus in quantum convolutional neural networks. *Physical Review X* 11, 4 (2021), 041011.
- [16] Chase Roberts et al. 2019. Tensornetwork: A library for physics and machine learning. *arXiv preprint arXiv:1905.01330* (2019).
- [17] Maria Schuld et al. 2019. Evaluating analytic gradients on quantum hardware. *Physical Review A* 99, 3 (Mar 2019). <https://doi.org/10.1103/physreva.99.032331>
- [18] Sukin Sim et al. 2019. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies* 2, 12 (2019), 1900070.
- [19] J.C. Spall. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Automat. Control* 37, 3 (1992), 332–341. <https://doi.org/10.1109/9.119632>
- [20] Yandell et al. 2002. Genomics and natural language processing. *Nature Reviews Genetics* 3, 8 (2002), 601–610.
- [21] Chen Zhao et al. 2021. Analyzing the barren plateau phenomenon in training quantum neural networks with the ZX-calculus. *Quantum* 5 (2021), 466.