Quantum Depth for Evaluating Transformer-Like Models

Ahmad Sohrabi (ahmad.sohrabi@carleton.ca)

Carleton University Cognitive Science Technical Report 2025-02

cogscitechreports@carleton.ca

Institute of Cognitive Science
2201 Dunton Tower
Carleton University
1125 Colonel By Drive
Ottawa, Ontario, K1S 5B6
Canada

Quantum Depth for Evaluating Transformer-Like Models

Ahmad Sohrabi (ahmad.sohrabi@carleton.ca)

Cognitive Science Department, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada K1S 5B6

Abstract

Quantum states are deep information representations especially with rotations and entanglement. We prepared a 2-qubit ansatz circuit to manipulate the quantum depth levels as a challenging minimal benchmark for assessing the scalability of a transformer-like model. For this purpose, a small dataset of state vectors using small angle rotations at 5 levels of depth was prepared to train and evaluate two model types, feed-forward networks with or without self-attention module, also varied in their parameter growth. This was done by reducing the number of hidden dimensions from 8 to 4 or the number of attention heads from 4 to 1. The performance was measured in terms of speed for learning to map inputs to targets. The simulation results showed a gradual performance drop as the depth level increased while hidden dimension and attention heads could compensate each other to some extent.

Introduction

While we can still feel spookiness in some aspects of quantum field theory at its 100th anniversary, its explanatory power is higher than ever. On the other side, we also continue to witness the great successes of recent attention–based neural networks, known as Large Language Models (LLM) or the Generative Pre–trained Transformers (GPT). Upon proving the scale-up benefits of such models, novel research works on explaining and improving their mechanisms and processes have been skyrocketing.

An ever-expanding horizon in this regard is the development of valid and challenging benchmarks for their assessment and validation. While traditional neural networks and deep learning AI models mainly focused on 'engineering' benchmarks such as Iris [1] and MNIST [2], current models have been assessed against a variety of benchmarks including but not limited to symbolic [3], for a review see [4], or non-symbolic reasoning, e.g., Abstraction and Reasoning Corpus, ARC, as described in [5].

However, most researchers accept that still issues remain to be solved, knowing the models' failures or shortcomings, for example in abstract reasoning [3-5] or planning tasks [6], while their successes continue [7]. Another issue is the limits of explanatory and visualization methods for LLM models compared to more conventional [8]. So, looking at the behavior and mechanism of these popular models in new domains merits further exploration.

Moreover, only a few research benchmarks exist with good discriminatory precision, focused on lightweight models, similar to those used for educational purposes including the well-known nanoGPT [9]. Existing benchmarks are mainly using texts (in the mentioned case, Shakespeare's works) or otherwise aimed at specific models such as convolutional neural networks [10]. Therefore, it seems useful having a research-oriented benchmark suitable for basic or abstract domains while allowing to play with parameters such as layers number, head number, embedding size, etc. Explanation and clarification of how the models work can enhance our understanding and help us to spot room for improvements and alternatives, while paving the road towards further advancement.

In this regard, the evaluation of these models on quantum data can have twofold benefits. It can be useful for their explanation and improvement while bringing new insights and methods to the quantum area, for example, by looking at how models are affected by different quantum state mappings. In the current study, we tested a minimal model on a small quantum information dataset. The quantum representation (state) involving complex valued numbers is among less explored areas to train and test these models on. Therefore, we prepared a dataset of 2-qubit-based state vectors in the form of complex numbers. Then, as will be described

below, they were used for training and testing a transformer encoder consisting of a Feed-Forward Network (FFN) module with or without a self-attention module (Figure 1). We explored the model's learning performance through scale-up with a gradual increase in hidden dimension size and the number of attention heads at different levels of quantum depth. While in quantum studies the main goal has been to find ways to reduce or optimize the circuit depth [11] and [12], i.e., as a dependent variable in methodological sense, here we are going to treat it as an independent variable for model testing, though it has implications for other purposes too as will be discussed later.

Quantum Information and Depth

Quantum information is commonly encoded as qubits and undergoes transformations mainly through unitary gate-based operations on multi-qubit circuits. The encoding or representation of information is realized in the form of state vectors from each arrangement or the so-called ansatz of gates and connections. While the circuit width is mainly aligned with the number of qubits and controlled and non-controlled connections, its length is determined by the serial iterations of gate operations, close to what is known as quantum circuit depth. The depth of a quantum circuit is simply based on its gate layout and temporal sequences and is an important rigorous topic [13], widely studied in quantum domains including hardware, simulations, and mathematical theorems [14] [15] [16] [17] [18].

Although depth is usually understood as the sequential execution of operations, it is difficult to define and implement precisely. This is because the depth is affected by the type and/or number of gate operations and connections between them, implementing rotations, entanglements, and other transformations. Thus, different terminologies and aspects are involved such as constant depth, linear depth, circuit depth, feature-map depth, finite depth, gate-aware depth and so on [13] [14] [16] [17]. On the other hand, the number of qubits despite being fixed is also included in circuit depth estimation, arguing they increase linearly together [13]. But in most studies, this is not taken into account because the operations are run in parallel, so the sequential length matters. As will be described later, we consider the depth as sequential gate operations that can contain entanglements and rotations, making the prepared quantum states challenging enough when processed by AI models.

Methods

The Model Architecture

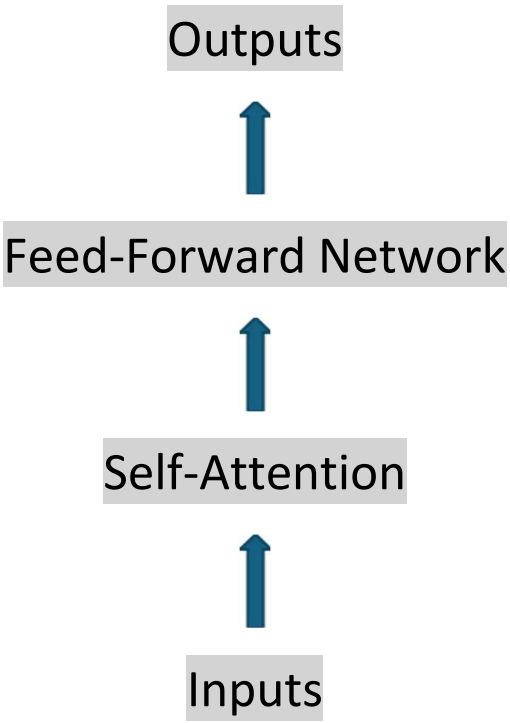


Figure 1. The model architecture with two modules: FFN and Self-Attention.

The model is a simplified version of transformer-like models [19] [20] [21]. The model was modified for quantum domain to deal with complex valued state vectors. It has two main modules, a simple linear FFN and a self-attention. The self-attention module uses no mask, and no value projection similar to [22]. The FFN module is a one linear layer using Rectified Linear Unit (ReLU) with no dropout. Also, it lacks embedding, layer normalization, and positional encoding. The popular Adam optimizer was used (for details and alternatives see [23]) for gradient-based training with an initial learning rate of 0.001 and a fidelity-based loss function (for details and alternatives see [24]).

This configuration is aimed to make a small model version; laser focused on a specific purpose to explore the roles of its two components and two main parameters in an experimental modeling approach. The model was scaled up by increasing two parameters: the number of heads in the self-attention module (1to 4) and the hidden dimension size (4 to 8), which connect the modules. Two versions of the model were studied. In one condition, an FFN module without a self-attention module (henceforth, FFN-Only), where the hidden dimension sizes varied, i.e., 4, 6, and 8. In the other condition, the model had both FFN and self-attention modules (henceforth, FFN + Attention), where the hidden dimension size was fixed at 4 but the self-attention module had varying head numbers, i.e., 1, 2, and 4 (with 0 attention head it will become an FFN-Only with 4 hidden dimensions). The model with its current minimalistic size and components can be seen as a quantum-oriented, simplified counterpart of the well-known

nanoGPT [9]. Though it is like GPT, especially the FFN + Attention, it does not apply recursive mapping in its conventional sense, i.e., not an auto-regressive generative model.

Moreover, working with quantum information as data benchmark to feed into a transformer-like model which is based on classical methods has its own complications, including common optimizers' focus on non-complex gradients. But since we only manipulate the depth, it allows us to specifically analyze the cross-section domain with the least involvements in quantum aspects. Still, we considered the required adoption of the main components of the model for complex valued implementation of the quantum side using recent pyTorch functions tailored for complex-valued [25]; sample codes can be found in [26]. In addition to the model's parameters, type and size, the main independent factor to explore here is the complexity depth with 1-5 levels, as is described below.

Quantum Data Preparation

The quantum dataset for training the model was constructed following an approach based on qubits. The dataset consists of input-target pairs representing two qubit quantum states encoded as state vectors having complex-valued amplitudes. All state preparations were made using IBM Qiskit ([27]; for a description see [28]; and for alternative methods see python functions in [26]). Each input and target state vector were derived from the same ansatz circuit (Figure 2, Top). A sample state vector using Dirac notation in an H_4 Hilbert space is shown in Figure 2(Middle) from final state at depth 4. They represent the superposition-based entanglement (partial Bell states) as also depicted with a Qsphere (Figure 2, Bottom).

Here, similar to Qiskit depth function and the textbooks' definitions ([29], page 18; see also their Figure 1.2), we simply operationalize the depth in terms of the repetition of the sequential gate operations in the whole ansatz in a circular order for a number of times (Figure 2, Top, notice the separation of slices/ layers of depth by a barrier). We call it complexity depth as it is based on a relational combination of circuit components assuming an idealized condition with no noise, errors, or other artifacts. On real hardware these and other issues arise, for example the transpilation and execution time differences per gate operations [17]. Since we use the same gates in each layer, the depth here can be imagined as number of layers in variational quantum algorithms [18].

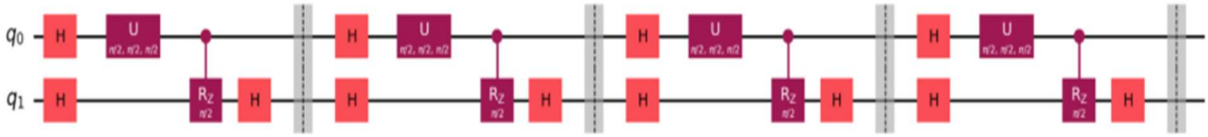
As shown in Figure 2, each 2-qubit circuit with a specific depth was created through non-Hermitian unitary operations, started with initial superpositions through Hadamard gates (H) on both qubits followed by a U gate with random angle parameters. This was also followed by an entanglement in the form of a controlled Z rotation (CRZ gate) to the second qubit as target with a random rotation angle, sandwiched between a final Hadamard gate and a previous one.

The U gate had angle parameters, randomly picked from an epsilon (π /360) to π, then multiplied by π /360 to minimize angle accumulation and entropy. This was followed by the controlled Z rotation (CRZ gate), with a rotation angle randomly picked from π /360 to π. These transformations were repeated up to the depth level for the Input and the Target separately but via the same circuit for each pair (see Figure 2 and more details below). This formed pairs with specified depths, for inputs ranging from 1 to 5 and targets from 2 to 10. For example, at level 2, the target keeps the input depth 2 and goes through two more levels. Therefore, a target at depth level 2 and an input at depth level 4 have the same depth (both have depth 4 as shown in Figure 2, Top). In the analysis, we encode them as 1-5 depth level though they might be different from other depth measures. If we use similar technical measures of depth as previously mentioned e.g., [11], the result will ultimately be a linear function of common measures, making no difference for our purpose.

Simulations on the Dataset

In each simulation run, one type of model was used with a given size and depth level. At the start of each run, a state vector was derived from the ansatz and was added to the dataset until a 360-sample dataset was generated. It for one simulation run out of 1000 for each of the 5 depths, 2 model types, and 3 model sizes, making the total simulations 30,000 runs overall. The angles for each rotation were different per gates, per Input/Target pairs, and per simulations. The batch size was 0.1 (36/360) and the Val/Test to Train ratio was 0.2 (72/ 360 with equal Val/Test length, 0.1/0.1).

As mentioned before, two types of models were used, an FFN-Only, with no self-attention, with hidden dimension sizes varied (4, 6, and 8, called size 1-3). For the other type (FFN + Attention), the hidden dimension size was fixed at 4 but using a self-attention module with varying head numbers from 1, 2, and 4 (i.e., size 1-3). The depth level was ranked from 1-5 as described previously.



$$|\psi\rangle = (-\frac{1}{4} + \frac{\sqrt{2}i}{2})|00\rangle + \frac{i}{4}|01\rangle + (\frac{\sqrt{2}}{4} + \frac{i}{4})|10\rangle + (-\frac{1}{4} - \frac{\sqrt{2}i}{4})|11\rangle$$

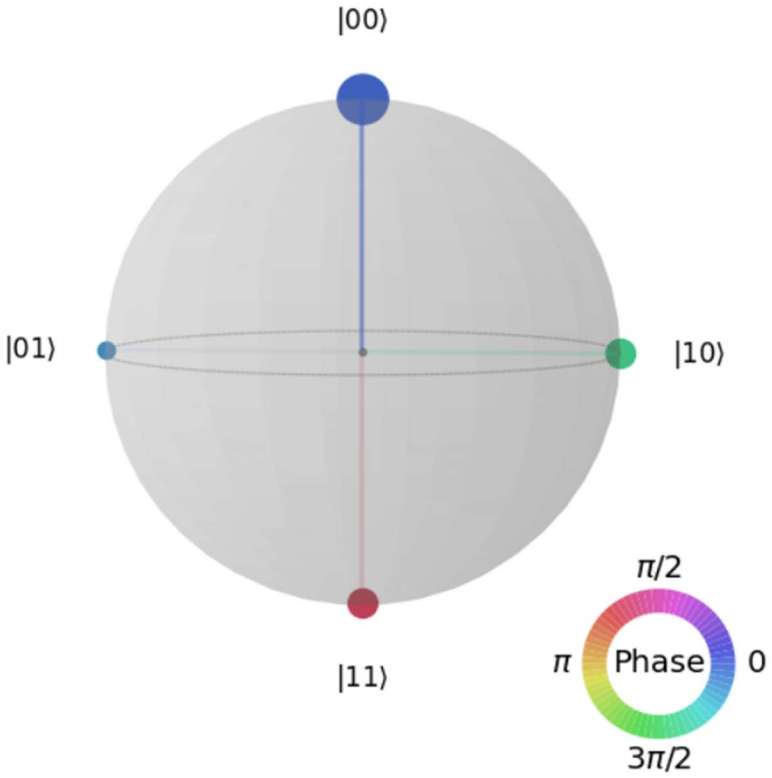


Figure 2. Top: An example ansatz of depth 4 where unitary operations (H and U) are applied on the first qubit, followed by an entanglement in the form of a controlled Z gate to the second qubit as target sandwiched between a final Hadamard and a previous one on the second qubit. Middle: The state vector derived from the ansatz in Dirac notation. Bottom: A visualization of superposition-based entanglement depicted with a so called Qsphere.

Results

The main factor considered in the analyses of each simulation was the number of epochs needed to reach 90 percent accuracy in both training and evaluation modes of the model. The accuracy was measured as the number of correct outputs in terms of their similarity to the target (fidelity >= 0.95) upon feeding the inputs into the model each time.

As shown with a regression line in Figure 3 (Top), both models types (FFN with or without Attention altogether) regardless of type and size (the number of resources, i.e., hidden dimensions in FFN-Only and the number of heads in FFN + Attention), showed a gradual

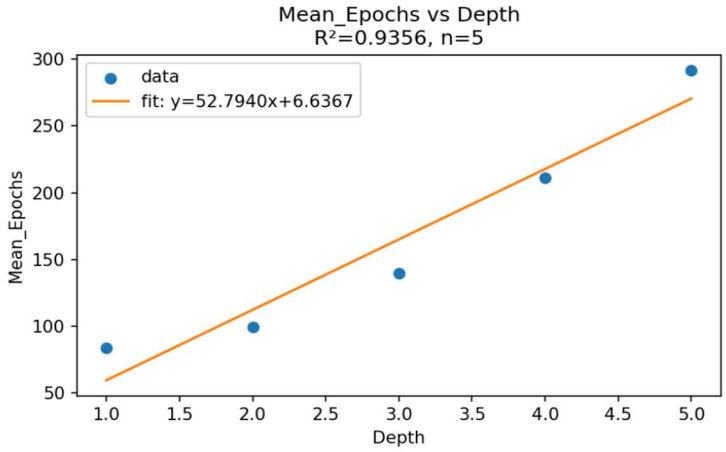
increase in the number of epochs needed to learn the map between inputs and targets as the depth level increased from 1 to 3 with an r^2 more than .90.

The same trend holds with slight differences when examining the detail separately for the FFN-Only (Figure 3, Middle) and FFN + Attention (Figure 3, Bottom). As shown in Figure 4 and Table 1, the FFN-Only exhibited a gradual increase in the number of epochs required to learn the input-target mappings as the depth level varied from 1 to 5. Likewise, size acted as an additive factor, as it decreased (from 8 to 4) the required epoch numbers increased accordingly. Additionally, the FFN + Attention showed a similar trend.

As shown in Figure 5 and Table 2, the FFN model with self-attention (namely FFN + Attention) showed a gradual increase in the number of epochs needed to learn the mappings as the depth level varied from 1 to 5, while the size also had an additive effect leading to an increase in the required epoch numbers as the number of attention heads decreased (from 4 to 1).

Therefore, comparing the two models in general shows an overall advantage for all model size measures, i.e., increasing the hidden dimensions from 4 to 8, when no attention was used and similarly by adding attention module and increasing the number of the attention heads from 1 to 4, while the hidden dimensions were fixed at 4. Further increase of the depth can lead to a stronger effect but also cause more instability as reflected in a linear trend of the Standard Error of the Mean (SEM) with the depth level (Table 1-2).

From Figures 4-5 (see also Tables 1-2) we can notice that there is a progressive expansion along the increase that shows an interaction of the depth and size effects. This means the effect of depth becomes more evident as model size decreases, particularly with FFN-Only.



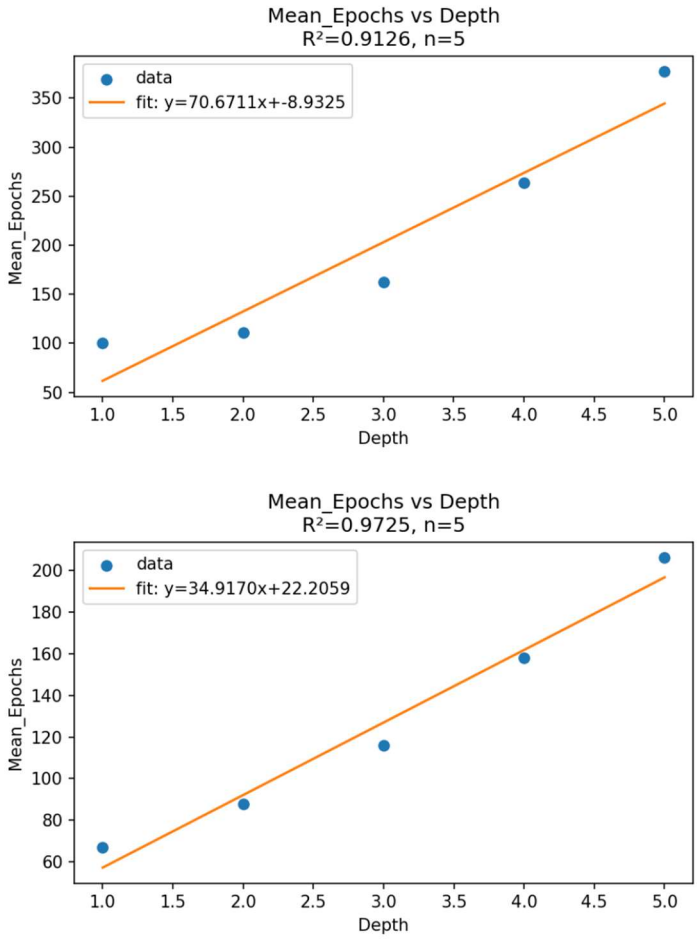


Figure 3. The correlation between depth and mean epoch numbers (required to reach a threshold). Top: The correlation of depth with mean epoch numbers, for both model types (FFN with or without Attention altogether) regardless of type and size. Middle: The correlation of depth with mean epoch numbers, for FFN-Only regardless of the model size. Bottom: The correlation of depth with mean epoch numbers, for FFN + Attention regardless of the model size.

Table 1. Results for FFN-Only condition. There is a gradual increase in the required epochs as the depth level varied from 1 to 5 and an increase in the required epochs as the number of attention heads decreased. Overall, in this condition, the required epoch numbers were higher than in FFN + Attention (Table 2).

Model_Type	Model_Size	Depth	Simulations	Mean_Epochs	SEM_Epochs
Model: FFN-Only	1	1	1000	141.446	2.427585
Model: FFN-Only	1	2	1000	154.029	2.336475
Model: FFN-Only	1	3	1000	227.894	3.325262
Model: FFN-Only	1	4	1000	417.623	7.567556
Model: FFN-Only	1	5	1000	639.606	15.407217
Model: FFN-Only	2	1	1000	92.506	1.326816
Model: FFN-Only	2	2	1000	100.740	1.264921
Model: FFN-Only	2	3	1000	148.379	1.838405
Model: FFN-Only	2	4	1000	220.204	3.136113
Model: FFN-Only	2	5	1000	297.961	6.011264
Model: FFN-Only	3	1	1000	67.917	1.078870
Model: FFN-Only	3	2	1000	77.190	0.931522
Model: FFN-Only	3	3	1000	112.072	1.265705
Model: FFN-Only	3	4	1000	154.420	1.771730
Model: FFN-Only	3	5	1000	194.224	2.335471

The model size effect at deeper compared to shallower depth along with the linear increase in variance both reflect the randomness method employed for angle rotations, starting from very small to large angles that with depth it becomes larger though its upper bound is π. As can be noticed, a decrease in the hidden dimension size for example to 6 or 4 without self-attention leads to substantial drops in performance and causes issues such as barren plateau and extreme variance.

Table 2. Results for FFN + Attention. There is a gradual increase in the required epochs as the depth level varied from 1 to 3 and an increase in the required epochs as the number of attention heads decreased. Overall, here the required epoch numbers were lower than in FFN-Only (Table 1).

Model_Type	Model_Size	Depth	Simulations	Mean_Epochs	SEM_Epochs
Model: FFN + Attention	1	1	1000	72.519	1.031466
Model: FFN + Attention	1	2	1000	98.629	1.211029
Model: FFN + Attention	1	3	1000	132.902	1.637115
Model: FFN + Attention	1	4	1000	182.581	2.190800
Model: FFN + Attention	1	5	1000	233.560	2.665901
Model: FFN + Attention	2	1	1000	65.426	0.808932
Model: FFN + Attention	2	2	1000	85.926	1.018315
Model: FFN + Attention	2	3	1000	112.695	1.436174
Model: FFN + Attention	2	4	1000	154.242	1.840880
Model: FFN + Attention	2	5	1000	205.895	2.553175
Model: FFN + Attention	3	1	1000	62.652	0.823158
Model: FFN + Attention	3	2	1000	78.798	0.914105
Model: FFN + Attention	3	3	1000	101.957	1.124590
Model: FFN + Attention	3	4	1000	136.820	1.532879
Model: FFN + Attention	3	5	1000	179.752	1.962997

Overall, the required epoch numbers in FFN-Only (Table 1) were higher than FFN + Attention (Table 2), indicating an advantage of adding self-attention, especially at deeper levels. As mentioned, ansatz circuits consist of rotation and entanglement can involve relative phase shifts, phase accumulation, and other complications. Regardless of these details, their effect on depth and model performance is pronounced considering the required epochs to reach a designated accuracy threshold (>= 0.90) in both train and evaluation stages.

Finally, although we had fixed accuracy threshold but still accuracy results were similar to required epoch measure where faster models passed the accuracy threshold earlier at either

one of train, eval, or test time.

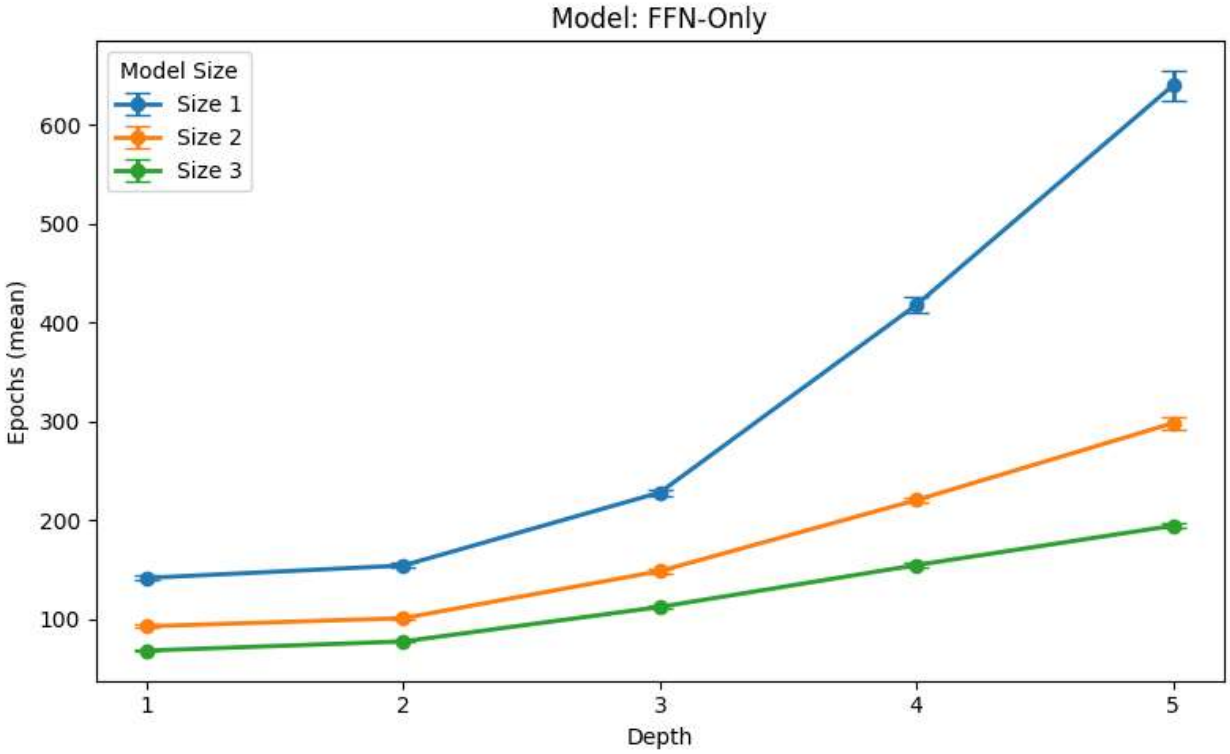


Figure 4. The FFN-Only performance: Mean epochs at five depth levels separately for the three model sizes.

Discussion

An important issue in AI modelling is the way knowledge is represented along with the way it is processed. Here we focused on this issue by feeding quantum state vectors with gradual depth directly into a transformer-like neural network model and observed the performance in terms of speed, measured by the required epochs to reach an accuracy threshold. The model's scale in terms of hidden dimension size and number of self-attention heads was correlated with performance while affected by the depth level. There was a gradual increase in the number of epochs needed to learn the map between inputs and targets as the depth level increased. This was revealed by a systematic reduction of either the hidden dimensions size from 8 to 4 or the number of attention heads from 4 to 1. Overall, deeper quantum information mappings required models with larger parameters, such as hidden dimensions size and/or the number of attention heads, while to some extent they could compensate each other, for example with 4 attention heads and still 4 hidden dimensions.

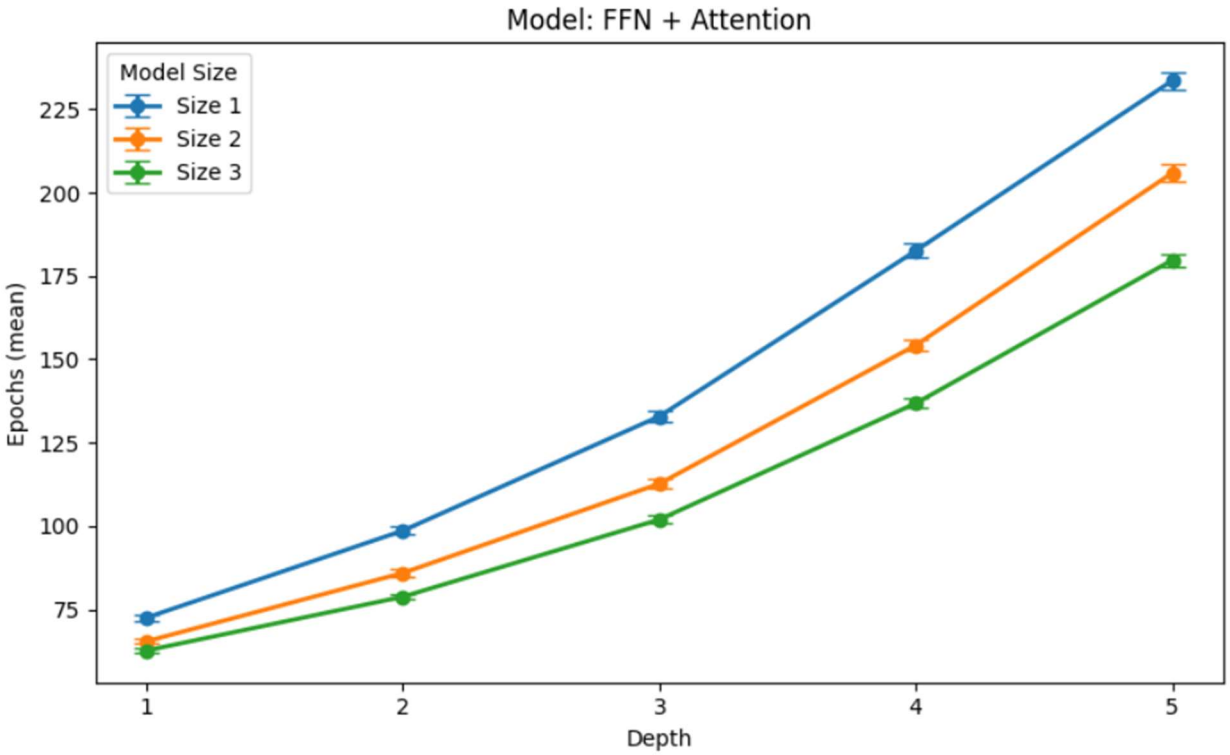


Figure 5. The FFN + Attention performance: Mean epochs at five depth levels separately for the three model sizes.

We obtained these results with a simple method to measure the quantum depth which is quite in line with known methods in this area. This depth benchmark helped to evaluate our simple transformer-like model, tailored for quantum data. Most studies in quantum computing have mainly focused on methods for decreasing and optimizing the depth [13] [16] [30] [31] or using it as a learning method not training data [18]. In the latter case, the so-called finite depth has been explored for studying learning algorithms. Furthermore, it is eventually similar to the structural and sequential layering depth count and correlates with our level of depth scale and linearly increases by each iteration.

However, depth optimization is still an ongoing important subject [13]. The entangling gates we used can make complicated issues such as phase accumulation and shifts similar to issues with CNOT [31]. Although the CZ is a stable gate compared to other control operations and is more complex than the linear CNOT, but with certain angles (i.e., π) it acts like a CNOT. This cannot be crucial for our study as [31] explored an optimization algorithm to reduce depth, particularly by eliminating the CNOT, unlike the nonlinear CZ we utilized with random small rotations.

The depth level exploration in the way we employed not only can enhance our understanding of classical AI models but also might help to validate quantum techniques such as cryptography

[32]. Similarly, it may inspire new development ideas for quantum algorithms like Shor's and Steene's [33], perhaps as another steppingstone toward quantum AI models or to combat their issues such as errors, gradient vanishing, barren plateau, and so on [34]. These techniques can be inspired mainly by the depth method with proper implementations in the form of memories or other cognitive processes models.

This was achieved with no embedding, tokenization, and layer normalization as all are in the data in the form of state vectors and further changes can alter them. However, manipulating the number of heads above four, i.e., more than the number of input quantum states without adding embedding (embed size > quantum states) and/or increasing the quantum states can be an issue because the number of heads will exceed the embed size, known as head/embed ratio rule. In all instances, the quantum states will be affected when the model size is increased, particularly the hidden dimension size. When hidden dimension size > 4, as we included for the FFN-Only model type, the state which is represented as real and imaginary numbers may start to get reduced to real numbers, similar to classical information. This is more evident with a better performance of the FFN + Attention than the FFN-Only model. But it might invoke similar issues, though we adopted linear layer and attention layer and other main processes to their complex valued versions, and added no dropout to avoid further alterations. So, the model was simplified without thorough parameterization because they are mainly related to the transformer models not quantum representations and are very broad with increasing diversities.

On the AI modeling side, looking further at the element and parameters that we included may deepen our understanding. These might be of interest in future studies, for example exploring the role of factors like residual/skip connections [35], optimizer [24], and loss functions [25]. Exploring new methods can also be interesting, such as enhancing attention heads through head gating [36] or adding positional encoding and position coupling [37]. Furthermore, using current data, we found out that some enhancement methods make no significant difference, like activation function GeLU compared to ReLU that we picked [38], or seems to be detrimental, such as dropout [39] as well as layer normalization even dynamic ones [40].

Our benchmark was made with small differences between input and target, making learning more like an autoencoder, especially at shallower depth. Mapping quite different pairs (with low fidelity) or in non-mapping settings will be more challenging for quantum representations but also requires more parametrization and optimization. However, autoencoding or auto-association is like error correction or stabilization, the dominant approach in the current mainstream quantum computing so may have implications for this area too. As mentioned, a recent study [14] has used a more direct measure of depth as the number of feature-map operations up to depth 2 and different from the repetitions of gate operations we used.

Therefore, investigating new measurements or formulations for depth is still much needed in future studies.

Acknowledgements

To be added.

Code Availability

The supplementary materials and sample codes will be provided in [26].

References

[1] Fisher, R.A.: The use of multiple measurements in taxonomic problems. Ann. Eugenics 7(2), 179–188 (1936). https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

[2] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE 86(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791

[3] Nezhurina, M., Cipolina-Kun, L., Cherti, M., Jitsev, J.: Alice in Wonderland: Simple Tasks Showing Complete Reasoning Breakdown in State-Of-the-Art Large Language Models. arXiv preprint arXiv:2406.02061 (2025). https://doi.org/10.48550/arXiv.2406.02061

[4] Ahn, J., Verma, R., Lou, R., Liu, D., Zhang, R., Yin, W.: Large Language Models for Mathematical Reasoning: Progresses and Challenges. In: Proc. EACL SRW 2024, pp. 225–237. https://doi.org/10.18653/v1/2024.eacl-srw.17

[5] Chollet, F.: On the measure of intelligence. arXiv preprint arXiv:1911.01547 (2019). https://doi.org/10.48550/arXiv.1911.01547

[6] Chen, X., Xu, J., Liang, T., He, Z., Pang, J., Yu, D., et al.: Do NOT Think That Much for 2+3=? On the Overthinking of o1-Like LLMs. arXiv preprint arXiv:2412.21187v2 (2025). https://doi.org/10.48550/arXiv.2412.21187

[7] Tian, Y., Peng, B., Song, L., Jin, L., Yu, D., Han, L., Mi, H., Yu, D.: Toward Self-Improvement of LLMs via Imagination, Searching, and Criticizing. In: *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*. https://openreview.net/pdf?id=tPdJ2qHkOB

[8] Kumar, A., Clune, J., Lehman, J., Stanley, K.O.: Questioning Representational Optimism in Deep Learning: The Fractured Entangled Representation Hypothesis. arXiv preprint arXiv:2505.11581 (2025). https://doi.org/10.48550/arXiv.2505.11581

[9] Karpathy, A.: NanoGPT. GitHub repository. https://github.com/karpathy/nanoGPT

[10] Su, Y.-C., Grauman, K.: Dataset: Spherical MNIST. LDM Dataset Repository. https://service.tib.eu/ldmservice/dataset/spherical-mnist

[11] Rambow, P., Tian, M.: Reduction of circuit depth by mapping qubit-based quantum gates to a qudit basis. arXiv preprint arXiv:2109.09902 (2022). https://doi.org/10.48550/arXiv.2109.09902

[12] Gyongyosi, L., Imre, S.: Circuit depth reduction for gate-model quantum computers. Sci. Rep. 10, 12290 (2020). https://doi.org/10.1038/s41598-020-67014-5

[13] Zi, W., Nie, J., Sun, X.: Constant-depth quantum circuits for arbitrary quantum state preparation via measurement and feedback. arXiv preprint arXiv:2503.16208 (2025). https://doi.org/10.48550/arXiv.2503.16208

[14] Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., Woerner, S.: The power of quantum neural networks. Nat. Comput. Sci. 1(6), 403–409 (2021). https://doi.org/10.1038/s43588-021-00084-1

[15] Zhang, X.-M., Li, T., Yuan, X.: Quantum state preparation with optimal circuit depth: Implementations and applications. Phys. Rev. Lett. 129(23), 230504 (2022). https://doi.org/10.1103/PhysRevLett.129.230504

[16] Yuan, P., Zhang, S.: Full characterization of the depth overhead for quantum circuit compilation with arbitrary qubit connectivity constraint. Quantum 9, 1757 (2025). https://doi.org/10.22331/q-2025-05-28-1757

[17] Tremba, M., Hovland, P., Liu, J.: Is Circuit Depth Accurate for Comparing Quantum Circuit Runtimes?, *arXiv preprint arXiv:2505.16908 (2025)* https://doi.org/10.48550/arXiv.2505.16908

[18] Bravo-Prieto, C., Lumbreras-Zarapico, J., Tagliacozzo, L., Latorre, J.I.: Scaling of variational quantum circuit depth for condensed matter systems. arXiv preprint arXiv:2002.06210 (2020). https://doi.org/10.48550/arXiv.2002.06210

[19] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014). https://doi.org/10.48550/arXiv.1409.0473

[20] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017). https://doi.org/10.48550/arXiv.1706.03762

[21] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. OpenAI (2018). https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[22] D'Amico, F., Negri, M.: Self-attention as an attractor network: transient memories without backpropagation. IEEE Workshop on Complexity in Engineering (COMPENG), Florence, Italy (2024). https://doi.org/10.1109/COMPENG60905.2024.10741429

[23] Aula, S.A., Rashid, T.A.: Foxtsage vs. Adam: Revolution or evolution in optimization? Cogn. Syst. Res. 92, Article 101373 (2025). https://doi.org/10.1016/j.cogsys.2025.101373

[24] Ma, H., Sun, Z., Dong, D., Chen, C., Rabitz, H.: Tomography of quantum states from structured measurements via quantum-aware transformer. arXiv preprint arXiv:2305.05433v3 (2025). https://doi.org/10.48550/arXiv.2305.05433

[25] Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J.F., et al.: Deep complex networks. arXiv preprint arXiv:1705.09792 (2018). https://doi.org/10.48550/arXiv.1705.09792

[26] A GitHub repo here upon publication

[27] IBM Quantum: IBM Quantum Cloud Platform. https://quantum.cloud.ibm.com

[28] Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C.J., Lishman, J., Gacon, J., et al.: Quantum computing with Qiskit. arXiv preprint arXiv:2405.08810 (2024). https://doi.org/10.48550/arXiv.2405.08810

[29] Kaye, P., Laflamme, R., Mosca, M.: Quantum algorithms and applications. Batista Lab, Yale University (2007). https://files.batistalab.com/teaching/attachments/chem584/Mosca.pdf

[30] Zhang, L., Liu, Y.: Quantum circuit optimization for linear transformations. Quantum Inf. Process. 24, Article 04896 (2025). https://doi.org/10.1007/s11128-025-04896-2

[31] Meng, F., Liu, Y., Wang, L., Zhou, W., Zhou, X.: Low-depth quantum approximate optimization algorithm for maximum likelihood detection in massive MIMO. *Quantum Inf. Process.* **24**, 294 (2025). https://doi.org/10.1007/s11128-025-04896-2

[32] Khaleel, F. A., Tawfeeq, S. K.: Implementation of a modified noise-free and noisy multistage quantum cryptography protocol using QISKIT, *Quantum Studies: Mathematics and Foundations*, vol. 11, (2024). http://doi.org/10.1007/s40509-024-00344-5

[33] Huang, S., Brown, K.R.: Between Shor and Steane: A unifying construction for measuring error syndromes. *arXiv preprint* arXiv:2012.15403v2 [quant-ph] (2021). https://arxiv.org/abs/2012.15403

[34] Kwak, Y., Yun, W.J., Jung, S., Kim, J.: Quantum Neural Networks: Concepts, Applications, and Challenges. In: *Proc. 2021 Twelfth Int. Conf. on Advanced Computing and Communication Technologies*. IEEE (2021). https://ieeexplore.ieee.org/document/101373

[35] Wen, J., Huang, Z., Cai, D., Qian, L.: Enhancing the expressivity of quantum neural networks with residual connections. *Commun. Phys.* **7**, 220 (2024). https://doi.org/10.1038/s42005-024-01719-1

[36] Nam, A.J., Yang, Y., Cohen, J.D., Conklin, H.C., Griffiths, T.L., Leslie, S.-J.: Causal Head Gating: A Framework for Interpreting Roles of Attention Heads in Transformers. *arXiv preprint* arXiv:2505.13737v1 [cs.AI] (2025). https://arxiv.org/abs/2505.13737

[37] Cho, H., Cha, J., Awasthi, P., Bhojanapalli, S., Gupta, A., Yun, C.: Position Coupling: Improving Length Generalization of Arithmetic Transformers Using Task Structure. *arXiv preprint* arXiv:2405.20671v2 [cs.LG] (2024). https://doi.org/10.48550/arXiv.2405.20671

[38] Horuz, C.C., Kasenbacher, G., Higuchi, S., Kairat, S., Stoltz, J., Pesl, M., Moser, B.A., Linse, C., Martinetz, T., Otte, S.: The Resurrection of the ReLU. *arXiv preprint* arXiv:2505.22074 [cs.LG] (2025). https://doi.org/10.48550/arXiv.2505.22074

[39] Li, Q., Ke, W.: Investigating the Synergistic Effects of Dropout and Residual Connections on Language Model Training. In: *Proc. 47th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '24)*, Washington, DC, USA, July 14–18, 2024. ACM, 6 pages. https://arxiv.org/abs/2410.01019

[40] Zhu, J., Chen, X., He, K., LeCun, Y., Liu, Z.: Transformers without Normalization. In: *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 14901–14911 (2024). https://arxiv.org/abs/2406.07629