Quantum Transformers For Image Generation

Atishay Narayanan
Department of Mathematics
Princeton University
Princeton, NJ, USA
atishay@princeton.edu

Niraj Babar
SenSIP Center, ECEE
Arizona State University
Tempe, AZ, USA
nbabar@asu.edu

Sunil Vittal
Department of Mathematics
Princeton University
Princeton, NJ, USA
sv2954@princeton.edu

Glen Uehara
SenSIP Center, ECEE
Arizona State University
Tempe, AZ, USA
guehara@asu.edu

Gennaro De Luca
School of Computing and Augmented Intelligence
Arizona State University
Tempe, AZ, USA
gennaro.deluca@asu.edu

Andreas Spanias
SenSIP Center, ECEE
Arizona State University
Tempe, AZ, USA
spanias@asu.edu

***Abstract*—Quantum computing is becoming increasingly relevant, in part due to its potential to revolutionize machine learning through parallel processing enabled by quantum mechanics principles, such as superposition and entanglement. Transformer models such as OpenAI's ChatGPT and Anthropic's Claude have gained much attention lately due to their versatility and availability. In this paper, we train Quantum-Classical hybrid transformer models for image generation tasks. We use two quantum attention algorithms in Generative Pretrained Transformer (GPT) models: QMSAN and QKSAN. We train the models on subsets of the MNIST digits dataset. In our research, we use Xanadu's PennyLane quantum SDK to implement our quantum algorithms and PyTorch to build our machine learning models. The challenges in our study included long computational times and large memory requirements for training quantum GPT (qGPT) models. Our results revealed that our proposed qGPT models achieve similar PSNR and lower SSIM scores compared to a classical image GPT.**

***Index Terms*—Quantum Machine Learning, Quantum neural networks, GPT, Transformers, QMSAN, QKSAN, Attention, Qubits, MNIST**

I. Introduction

Quantum computing has the potential to rapidly increase computation speed by enabling parallelism brought on by superposition and entanglement [1]. This is especially useful for large neural networks, which often have high computational complexity in the training stage. Thus, Quantum Machine Learning (QML) shows promise in demonstrating advantages over classical machine learning approaches. For image generation tasks, previous research has shown that there is a quantum advantage in generating images with quantum models [2].

However, it is important to note that quantum computing hardware has not yet reached the level of maturity needed to reap these benefits. Quantum hardware is still in the Noisy Intermediate-Scale Quantum (NISQ) era [3], meaning devices at the state of the art are plagued with noise sources that affect qubit measurement and ultimately results.

Transformer models, first introduced by Vaswani et al. [4], employ a self-attention mechanism that measures the relationships between all input tokens with each other. This allows them to capture nuances within data more effectively than recurrent or convolutional architectures, while being more efficient in training. The use of transformer models has led to state-of-the-art results in a wide variety of Natural Language Processing (NLP) tasks, such as translation and text summarization. In OpenAI's Image GPT [5], pixels are treated as tokens, allowing the transformer to learn the semantic structure of images. After giving the model a pixel sequence of a partial image, the model predicts the rest of the pixel sequence. Image GPT's success in image generation motivates our exploration of quantum attention mechanisms for generative image modeling.

In this paper, we propose a hybrid architecture by constructing an Image GPT that features a quantum attention mechanism in place of Scaled Dot-Product Attention [4]. In classical Multi-Head attention, attention scores are calculated via

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{h}_1,\ldots,\mathrm{h}_h)\,W^O, \quad (1)$$

where

$$\mathrm{h}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V), \quad (2)$$

and

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\big(\tfrac{QK^\top}{\sqrt{d_k}}\big)V. \quad (3)$$

Instead of projecting Query and Key matrices from the input tokens for each head, as in classical Multi-Head Attention, we individually embed vectors to the quantum domain from the input tokens using an amplitude encoding, and then get our attention scores using quantum methods. Then, we finish the process classically with a linear projection to obtain the Value matrix and perform matrix multiplication accordingly. The rest of the transformer follows the classical approach with a Multi-Layer Perceptron (MLP) Feed-Forward layer as seen in Figure 1. To quantitatively measure our model performance, we use metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [6].

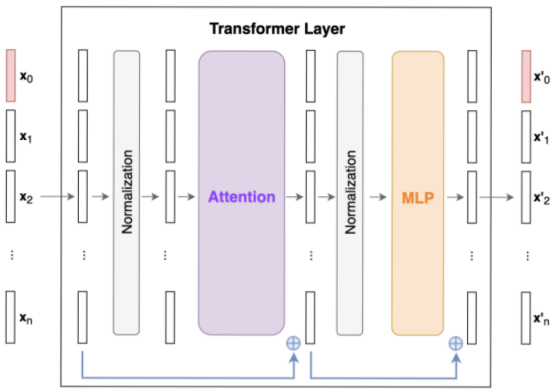

Fig. 1. Transformer Layer [7]

The remainder of this paper is organized as follows. In Section 2, we provide an overview of the architectures of our quantum Image GPT models and their respective attention mechanisms. In Section 3, we present our methods and results. In Section 4, we present our conclusions on the results of our simulations.

II. Proposed qGPT Architecture

A. Configuration

We propose two different quantum Generative Pretrained Transformer (qGPT) models: one with a Quantum Mixed-State Attention Network (QMSAN) [8] and one with a Quantum Kernel Self-Attention Network (QKSAN) [9]. The models have the hyperparameters seen in Figure 2. The classical Image GPT has 6,560 trainable parameters, while the qGPT with QMSAN has 4,880 trainable parameters, and the qGPT with QKSAN has 5,008 trainable parameters. The quantum models have fewer trainable parameters since they do not project the input tokens down the dimensions of each attention head; instead, they use an amplitude encoding scheme [10], which not only reduces parameters but also preserves information that is traditionally lost in the classical projection. Amplitude encoding allows us to represent $x \in \mathbb{R}^E$ in $\lceil \log_2 E \rceil$ qubits.

Hyperparameter	Classical	QMSAN	QKSAN
Vocabulary Size	16	16	16
Embedding Dimension	16	16	16
# of Attention Heads	4	4	4
Transformer Layers	2	2	2
Circuit Repetitions	N/A	N/A	1

Fig. 2. Model Hyperparameters

B. Preprocessing

Image GPT models use pixels as tokens and then predict the next pixel given a sequence of pixels. The pixels that the model can choose from for its predictions are fixed, so the vocabulary must be chosen carefully to faithfully represent the dataset that the model is trained on. To determine which pixels should be used as the model's vocabulary, we run Scikit-learn's K-Means [11] algorithm across each pixel in the entire dataset to get the "centroid" pixel values for a fixed vocabulary size. Since the model in this paper was trained on grayscale digit images, we determined that only 16 centroids were needed to adequately express the nuances of the data. However, it is important to note that, for more complex grayscale images or images with RGB pixels in three channels, significantly more centroids are needed to accurately express the dataset.

After we calculate the centroids, we then replace each pixel in the dataset with its nearest centroid pixel by Euclidean distance. In practice, we quantize each pixel by the index of its centroid for efficiency. Since we train the model on these quantized images, it is important to quantize any of the context sequences that we use for prediction.

C. QMSAN

QMSAN replaces the scaled dot-product with a quantum similarity measure between query and key embeddings [8]. Each token embedding $x \in \mathbb{R}^E$ is amplitude-encoded into a pure state $|x\rangle$ on $\lceil \log_2 E \rceil$ qubits. This is done twice to represent a query-key pair. Unfortunately, we were unable to implement a parameterized circuit after this due to the limitations of simulating circuits with mixed states. We then perform a SWAP test [12] over the first half of the qubits for each query-key pair, which effectively performs the SWAP over the equivalent mixed states $\rho_q\ \sigma_k$. Measuring the ancilla yields

$$p(|0\rangle) = \frac{1}{2} + \frac{1}{2}\mathrm{tr}(\rho_q \sigma_k). \tag{4}$$

Scaling by 2 and subtracting 1, we set each attention score to

$$\alpha_{s,j} = \mathrm{tr}(\rho_{s.q}\sigma_{j,k}). \tag{5}$$

Since $\alpha_{s,j} \in [0, 1]$, there is no need to scale by $\frac{1}{\sqrt{d_k}}$, and performing a Softmax not have the intended effect. Instead, we perform a simple normalization before we multiply by our V matrix, which is projected classically from the input tokens. Each head of attention is then concatenated, and this undergoes another linear transformation as in (1).

This method obeys the laws of quantum mechanics. However, it runs $O(S^2)$ SWAP circuits, where S is the sequence length, per attention head and requires 9 qubits; 4 per query-key vectors and 1 ancilla. In an alternative method, we precomputed the mixed state for each query and key vector individually, and computed each $\mathrm{tr}(\rho_q\sigma_k)$ with the saved mixed states. This method is mathematically equivalent and runs $O(S)$ quantum circuits while only requiring 4 qubits. However, this would not work on quantum hardware and is thus considered a quantum-inspired [13] algorithm. When training our model, we preferred this method for its efficiency.

D. QKSAN

We were unable to realize the full advantage of QMSAN since we could not implement a trainable embedding for our query-key pairs due to the limitations of simulating mixed states at the moment. This motivated us to pursue an alternative

to mixed states. Inspired by QKSAN [9], we decided to measure the similarity between the pure states of encoded query-key pairs. Since these states are essentially probability distributions, we decided on Jensen-Shannon Divergence [14] for our similarity measure. For distributions P and Q, the Jensen-Shannon Divergence is given by

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2}\mathrm{D}(P \parallel M) + \frac{1}{2}\mathrm{D}(Q \parallel M). \tag{6}$$

Here, $M = \frac{1}{2}(P + Q)$ and D represents the Kullback-Leibler Divergence [15] given by

$$\mathrm{D}(P \parallel Q) = \sum_{x \in \chi} P(x) \log \frac{P(x)}{Q(x)}. \tag{7}$$

In Kullback-Leibler Divergence, P is assumed to be the ground truth distribution.

Since we no longer needed to use mixed-state simulators, we were able to implement a trainable quantum circuit in this self-attention mechanism. We chose Circuit 14 from [16], seen in Figure 3, which was noted for its high expressibility. Using the same technique as before, we saved the pure states for each query and key vector after running it through our learnable embedding, and then computed the JS divergence to use as our attention scores. From there, we multiply by V and replicate (1) as in the QMSAN version. Similarly, this runs $O(S)$ quantum circuits while requiring 4 qubits.

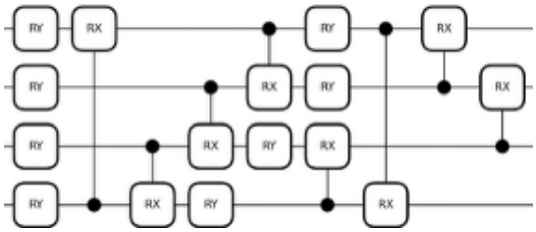

Fig. 3. Circuit 14 from [16] drawn with PennyLane and Matplotlib

III. Implementation and Results

A. Hardware Specs and Training Process

The simulations described below were performed on an Apple MacBook Pro with an Apple M3 CPU, 32 GB RAM, and macOS Sequoia 15.5. Moreover, circuits and quantum neural networks were developed using PennyLane [17] SDK, while the framework and classical components of the neural networks were developed using PyTorch. We compare and contrast three models: the qGPT with QMSAN, the qGPT with QKSAN, and a classical image GPT. Each model was trained on 5000 10×10 zero and one-digit images that were downsampled from the MNIST dataset. The models were trained for 8 epochs each, using ADAM [18] as the optimizer with a learning rate of 0.0003.

These models are evaluated qualitatively by examining the completions of the images and quantitatively by examining metrics including PSNR, SSIM, and training time. For our models, PSNR and SSIM were measured using the scikit-image library.

B. Qualitative Evaluation

In the classical Image GPT, we see that while the shape of the completions for zero digits are adequate, the model struggles to get a straight line for the one digits. As seen in Figure 4, the model also struggles to adequately predict the intensity of several pixels, especially in the top-right completion. The QMSAN struggled to accurately complete the inside of the zero shape, but did well to round out the outer silhouette in ways the other models could not. While some of the one digits it produced were good, they lacked the correct pixel intensity predictions. The QKSAN lacked in its zero digit predictions, especially with the shape. There were lackluster results for the one digits. However, it did well in predicting the correct pixel intensities for both classes, something the models struggled with. It is important not to overgeneralize from these results, as the dataset had to be trimmed down and the images had to be reduced in size to be able to train the quantum models within a reasonable amount of time. Thus, it is difficult to accurately identify the nuances of each model through qualitative means.

Fig. 4. Images generated by the classical, QMSAN, and QKSAN models, respectively. The leftmost column is the context sequence given to the model, the middle three columns are the model predictions, and the rightmost column is the ground truth.

C. Quantitative Evaluation

The results for each model based on PSNR and SSIM can be seen in Figure 5. As could be expected based on the images generated, the classical model outperformed both of the qGPT models in PSNR and SSIM. Between the QMSAN and QKSAN models, the QKSAN model outperformed in PSNR, while the QMSAN model exhibited a higher SSIM score. This nuance could be attributed to variance, but it is clear that both qGPT models lack in quality compared to the classical version. Despite this, they managed to stay within roughly 10% of the PSNR and SSIM scores.

What couldn't be seen in the images is the significant advantage that the classical model has in training time. The QMSAN model is quite slow, but it runs under an hour per epoch. On the other hand, the QKSAN model takes over 26 hours per epoch due to the high complexity of calculating gradients for quantum circuits. This exemplifies one of the main challenges faced by trainable quantum circuits in QML models: high computational resource requirements and high training times. While increasing the number of repetitions of the parameterized circuit in the QKSAN qGPT may have led to better images, doing so would have put further strain on resources, placing this outside the scope of this project.

Metric	Classical	QMSAN	QKSAN
PSNR	15.9674	14.6171	15.0445
SSIM	0.7449	0.7144	0.6339
Training Time	0.05	57.51	1572.63

Fig. 5. Average model scores after 8 epochs of training. Training time is measured in minutes per epoch.

IV. Conclusion

Although the quantum hybrid models did not outperform the classical Image GPT by any metric, the gap was relatively narrow in popular metrics such as PSNR and SSIM. Furthermore, the qGPT models were able to demonstrate the feasibility of using quantum attention algorithms in the place of dot-product attention within transformers. Notably, the QKSAN model performed comparably in PSNR and was able to implement a trainable quantum circuit, highlighting the potential of parameterized quantum attention layers. Unfortunately, both models suffered from high training times and resource requirements, highlighting the inefficiency of current quantum simulation capability. Looking ahead, many aspects of these models can and will be improved as research progresses. Additional future work includes training models with multiple parameterized circuit repetitions, moving beyond a 10×10 resolution, and training models on NISQ hardware.

V. Acknowledgments

This work has been sponsored in part by NSF CISE REU Project Award 2349567 and by the ASU SenSIP center.

References

[1] G. Uehara, A. Spanias, W. Clark, "Quantum Information Processing Algorithms with Emphasis on Machine Learning," Proc. IEEE IIS 2021, July 2021.
[2] S. Vittal, D. Ramirez, G. Uehara, and G. De Luca, Quantum Generative Neural Networks for Imaging Applications. 2024
[3] Preskill, John. "Quantum computing in the NISQ era and beyond." Quantum 2 (2018): 79.
[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.
[5] Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I., 2020, November. Generative pretraining from pixels. In International Conference on machine learning (pp. 1691-1703). PMLR.
[6] Nilsson, J. and Akenine-Möller, T., 2020. Understanding SSIM. arXiv preprint arXiv:2006.13846.
[7] Cherrat, et al., 2022. Quantum vision transformers. arXiv preprint arXiv:2209.08167.
[8] Chen, F., Zhao, Q., Feng, L., Chen, C., Lin, Y. and Lin, J., 2025. Quantum mixed-state self-attention network. Neural Networks, 185, p.107123.
[9] Zhao, R.X., Shi, J. and Li, X., 2024. Qksan: A quantum kernel self-attention network. IEEE Transactions on Pattern Analysis and Machine Intelligence.
[10] Weigold, M., Barzen, J., Leymann, F. and Salm, M., 2020, October. Data encoding patterns for quantum computing. In Proceedings of the 27th conference on pattern languages of programs (pp. 1-11).
[11] Ahmed, M., Seraj, R. and Islam, S.M.S., 2020. The k-means algorithm: A comprehensive survey and performance evaluation. Electronics, 9(8), p.1295.
[12] Zhao, J., Zhang, Y.H., Shao, C.P., Wu, Y.C., Guo, G.C. and Guo, G.P., 2019. Building quantum neural networks based on a swap test. Physical Review A, 100(1), p.012334.
[13] Moore, M. and Narayanan, A., 1995. Quantum-inspired computing. Dept. Comput. Sci., Univ. Exeter, Exeter, UK.
[14] Menéndez, M.L., Pardo, J.A., Pardo, L. and Pardo, M.D.C., 1997. The Jensen-Shannon divergence. Journal of the Franklin Institute, 334(2), pp.307-318.
[15] Van Erven, T. and Harremos, P., 2014. Rényi divergence and Kullback-Leibler divergence. IEEE Transactions on Information Theory, 60(7), pp.3797-3820.
[16] Sim, S., Johnson, P.D. and Aspuru-Guzik, A., 2019. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Advanced Quantum Technologies, 2(12), p.1900070.
[17] V. Bergholm et al, "PennyLane: Automatic differentiation of hybrid quantum-classical computations," 2018. arXiv:1811.04968.
[18] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.