Quantum Computing based Generative Adversarial Network for Time-Series forecasting

Aadittya Tiwari1, Bradley S Fulk1,2†

1,2 Lynbrook High School, 1280 Johnson Ave, San Jose, 95129, California

\*Corresponding author; E-mail: AadiTiwari1223@gmail.com   
†Project Mentor; E-mail: [brad\_fulk@fuhsd.org](mailto:brad_fulk@fuhsd.org)

Abstract

Generative Adversarial Networks (GANs) are a popular machine learning model that can generate synthetic data by training a generator to produce samples that a discriminator can't distinguish from real data. GANs have shown promise for time-series forecasting, where synthetic values can be used to predict future trends, such as stock prices. However, this can be taken one step further with the usage of Quantum Computers. In this project, I developed a novel Quantum GAN (QGAN) architecture that combined a Quantum variation of a Long Short Term Memory generator with a classical Convolutional Neural Network discriminator. The QGAN outperformed its classical counterpart in number of convergence epochs required and had a slightly higher prediction accuracy across a range of stock datasets, despite using a low qubit count. These results provide evidence of quantum supremacy in the domain of deep learning.

**Keywords:** Quantum Computing, Qubits, Machine Learning, Time-series Forecasting

# Introduction

# 1.2 Introduction to Quantum Computing

Quantum Machine Learning (QML) is an emerging interdisciplinary field combining quantum computing with machine learning algorithms to enhance data processing. Quantum Computers utilize “qubits” (quantum-bits) which exist simultaneously in multiple states, unlike classical bits that can only exist as either a 0 or 1, called quantum superposition. By leveraging the power to be in multiple states at once and perform parallel calculations faster through this ability, machine learning algorithms can be optimized for Quantum Computers to process and analyze vast amounts of data faster and obtain more precise results. As QML develops as a field, new algorithms are found frequently that demonstrate “Quantum Supremacy,” cases where Quantum Computing algorithms have advantages over Classical Computing algorithms. QML has the potential to transform research fields such as disease detection, physical simulations, but for the context of this project, time-series forecasting, by providing more accurate and faster solutions for these complex problems. This paper describes a novel quantum machine learning model that does indeed demonstrate Quantum Supremacy in the field of Generative Adversarial Networks.

# Quantum Computing and Machine Learning

## Simple introduction to Quantum Computing

Quantum computing is an emerging field of computer science that promises to revolutionize how we solve complex problems. Unlike classical computers, which rely on bits that can only exist in one of two states, quantum computers use quantum bits, or qubits, which can exist in multiple states simultaneously. This allows quantum computers to perform certain calculations much faster than classical computers, making them particularly well-suited for tasks such as optimization, cryptography, and in our paper, machine learning.

One of the key features of quantum computing which makes it so powerful is the phenomenon of superposition, which allows a qubit to exist in a combination of both 0 and 1 at the same time. In pop culture it's often said that an n number of qubits correlate to the power of 2^n classical its. While this isn't entirely true, it gives a rough idea of the computational advantage quantum computers provide. Another important feature is entanglement, where two or more qubits can be correlated in such a way that their states are linked together so that affecting one qubit will affect its entangled qubit as well. This paired with superposition is what provides such powerful exponential speed-ups in time complexities.

Quantum computing is still in its early stages, and many of the practical applications of this technology have yet to be realized. However, there has been significant progress in recent years, with the development of increasingly powerful quantum computers and the occasional demonstration and realization of quantum supremacy over classical computers for cherry-picked scenarios. However, Quantum Machine Learning as of late has been receiving a lot of attention in the last few years due to it being a conjunction of 2 of the most revolutionizing fields in Computer Science - Quantum Computing, and Machine Learning.

## Simple introduction to Machine Learning and GANs

## Machine learning is a branch of artificial intelligence that allows machines to learn from data without being explicitly programmed. It involves training algorithms to identify patterns in data and make predictions or classifications based on those patterns. The process of machine learning consists of three main stages: data preparation, model training, and model evaluation. During the data preparation stage, data is collected, cleaned, and transformed into a format suitable for analysis. In the model training stage, machine learning algorithms are used to train models on the prepared data. Finally, in the model evaluation stage, the trained models are tested on new data to ensure their accuracy and effectiveness.

**2.3.2** GANs, or Generative Adversarial Networks, are a type of neural network machine learning architecture that can generate new data that is similar to the training dataset. They consist of two models - a Generator and a Discriminator - that compete to learn and generate complex data such as images, audio, and video files. The Generator creates fake data to train on the Discriminator, which in turn learns to identify real data from the fake data produced by the Generator. This adversarial game between the two models continues until the Generator produces data that is indistinguishable from the real data.

# Q-GAN Model Architecture

Include equations, graphs, data tables, etc go ham with visuals. Have to cite images if they come from a different paper, same with equations. What did you replace, what does this particularly improve. How many qubits, how many this, how many that, why this, why that. Much more of the theoretical section.

Information flow in a quantum LSTM:

where denotes the sigmoid function, are classical neural networks , where represents the forget block, represents the input block, represents a new state cell candidate, and represents the output block.

Information flow in a quantum LSTM:

Where denotes a different Variational Quantum Circuit that will be used in the hybrid Quantum LSTM.

Given the functions above, we’re able to say that a QLSTM will outperform a LSTM in the field of...Switching the LSTM generator in a GAN for a QLSTM should therefore provide an advantage of...and advantage of... over just using an LSTM without a CNN discriminator.

# Actual Q-GAN and its code and stuff like that

Moving on from the technical architecture, we begin to implement the Quantum LSTM into a time series generator. Essentially, all we do is replace the classical generator in the GAN (which is currently a classical prebuilt PyTorch LSTM) with our custom Quantum LSTM. However, as our model is not prebuilt and the entire GAN relies on a prebuilt model, we will need to still keep the code for the prebuilt generator and instead create what is basically a second instance of the generator and call that instead, while the classical generator remains but not for the actual purpose of generation, more as a utility function.

Code for the classical GAN generator

Code for the quantum GAN generator

Discriminator which is the same for both

## Variables

The variables for both models were set after lots of experimentation to find optimal values to allow both functions to perform at their best but keeping as many values alike as possible to allow objective testing. The data table as follows shows what each value was set to in the following categories for the generators and the discriminators:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Generator | Learning Rate | Epochs | Optimizer | Layers |
| QGAN | 0.000016 | 15 | Adam | 4 |
| CGAN | 0.004 | 100 | Adam | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Discriminator | Learning Rate | Epochs | Optimizer | Layers |
| QGAN | 0.00003 | 300 | Adam | 3 |
| GAN | 0.00003 | 300 | Adam | 3 |

As you can see, the dimensions for the discriminator are identical since we are using the exact same Convolutional Neural Network (CNN) model for both GANs.

# Model evaluation methods (Experiments & Results)

## Evaluation Methods

We plan to evaluate the efficiency of the QGAN through the following 3 steps:

a) Selecting 3 stocks with differing trends to encourage robust models

b) Using an existing classical GAN for time series forecasting

c) Developing a novel QGAN model that changes ONLY the generator of the classical GAN

d) Evaluating the performance of both models on the 3 different stocks in the categories of prediction RMSE, epochs to converge, and parameters across all 6 predictions.

## Results

The following data table shows the performance of each model across the evaluation metrics defined above:

|  |  |  |
| --- | --- | --- |
| RMSE | CGAN | QGAN |
| Intel prediction | 7.11 | 13.10 |
| Apple prediction | 7.97 | 3.61 |
| Synopsys prediction | 4.77 | 8.50 |

|  |  |
| --- | --- |
| Epochs taken to reach convergence |  |
| QGAN | 6 out of 15 total epochs |
| CGAN | 80 out of 100 total epochs |

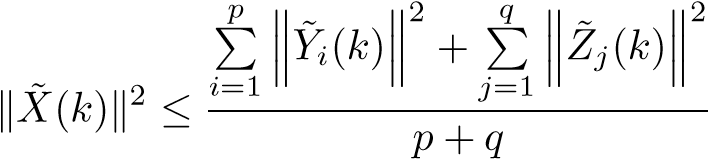
|  |  |
| --- | --- |
| Total parameters (Generator + Discriminator) |  |
| QGAN | 233+6199041 |
| CGAN | 109429+6199041 |

# Conclusion and Outlook

QML almost always returned higher accuracy for forecasts, albeit minimal. This higher accuracy is indeed a result of the QLSTM being used and having it perform calculations through a quantum computing simulator which then gave the QLSTM an advantage in converging quicker. QML was also able to reach comparable performance to CML in a tenth of the epochs. This is related to the Qubits above again - as Quantum Bits can exist in multiple states at once, it’s able to compute different possibilities at the same time. This means it trains more in one epoch than Classical ML does in one epoch because of these parallel time computations. Another interesting result was that there were significantly less parameters in the QML models, again as a result of having qubits that make up for the lack of parameters. Finally, as a result of using a simulator (a classical computer coded to behave like a quantum computer) and not a real quantum computer, it will actually take you more time to train your models unless you use a real computer. Possible room for improvement would be to try out different QC simulators from different companies, or in the far future evaluate performance on a real quantum computer. As hardware scales with time, so will algorithms making QML models such as this one far more effective and pave the way for algorithms that aren’t constructed from existing classical machine learning algorithms and are instead unique with no classical counterparts. Overall, the QGAN has shown itself to be a competitive counterpart to classical GAN’s in terms of performance and further shows the future of Quantum Machine Learning as a whole.

Equations in LATEX can either be inline or on-a-line by itself (“display equations”). For inline equations use the $...$ commands. E.g.: The equation *Hψ* = *Eψ* is written via the command $H\psi=E\psi$.

For display equations (with auto generated equation numbers) one can use the equation or align environments:

*.* (1)

where,

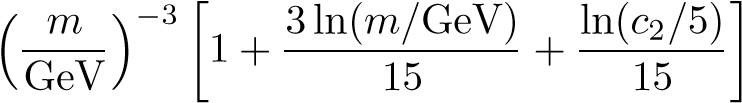
*λa a*

*Dμ* = *∂μ* − *ig**Aμ*

2

*Fμνa* = *∂μAaν* − *∂νAaμ* + *gfabcAbμAaν* (2)

Notice the use of \nonumber in the align environment at the end of each line, except the last, so as not to produce equation numbers on lines where no equation numbers are required. The \label{} command should only be used at the last line of an align environment where \nonumber is not used.

*Y*∞ =  (3)

The class file also supports the use of \mathbb{}, \mathscr{} and \mathcal{} commands. As such \mathbb{R}, \mathscr{R} and \mathcal{R} produces R, *R* and R respectively (refer Subsubsection 3.1.1).

# Tables

Tables can be inserted via the normal table and tabular environment. To put footnotes inside tables you should use \footnotetext[]{...} tag. The footnote appears just below the table itself (refer Tables 1 and 2). For the corresponding footnotemark use \footnotemark[...]

**Table 1** Caption text

|  |  |  |  |
| --- | --- | --- | --- |
| Column 1 | Column 2 | Column 3 | Column 4 |
| row 1 | data 1 | data 2 | data 3 |
| row 2 | data 4 | data 51 | data 6 |
| row 3 | data 7 | data 8 | data 92 |

Source: This is an example of table footnote. This is an example of table footnote.

1Example for a first table footnote. This is an example of table footnote.

2Example for a second table footnote. This is an example of table footnote.

The input format for the above table is as follows:

\begin{table}[<placement-specifier>]

\caption{<table-caption>}\label{<table-label>}%

\begin{tabular}{@{}llll@{}}

\toprule

Column1&Column2&Column3&Column4\\

\midrule

row1&data1&data2 &data3\\ row2&data4&data5\footnotemark[1]&data6\\ row3&data7&data8 &data9\footnotemark[2]\\

\botrule

\end{tabular}

\footnotetext{Source:Thisisanexampleoftablefootnote.

Thisisanexampleoftablefootnote.}

\footnotetext[1]{Exampleforafirsttablefootnote.

Thisisanexampleoftablefootnote.}

\footnotetext[2]{Exampleforasecondtablefootnote.

Thisisanexampleoftablefootnote.}

\end{table}

**Table 2** Example of a lengthy table which is set to full textwidth



Element 11 Element 22



Element 3 990 A 1168 1547 ± 12 780 A 1166 1239 ± 100 Element 4 500 A 961 922 ± 10 900 A 1268 1092 ± 40

Note: This is an example of table footnote. This is an example of table footnote this is an example of table footnote this is an example of table footnote this is an example of table footnote.

1Example for a first table footnote.

2Example for a second table footnote.

In case of double column layout, tables which do not fit in single column width should be set to full text width. For this, you need to use \begin{table\*} ... \end{table\*} instead of \begin{table} ... \end{table} environment. Lengthy tables which do not fit in textwidth should be set as rotated table. For this, you need to use \begin{sidewaystable}...\end{sidewaystable} instead of \begin{table\*} ...\end{table\*} environment. This environment puts tables rotated to single column width. For tables rotated to double column width, use \begin{sidewaystable\*} ...\end{sidewaystable\*}.

# Figures

As per the LATEX standards you need to use eps images for LATEX compilation and pdf/jpg/png images for PDFLaTeX compilation. This is one of the major difference between LATEX and PDFLaTeX. Each image should be from a single input .eps/vector image file. Avoid using subfigures. The command for inserting images for LATEX and PDFLaTeX can be generalized. The package used to insert images in LaTeX/PDFLaTeX is the graphicx package. Figures can be inserted via the normal figure environment as shown in the below example:

\begin{figure}[<placement-specifier>]

\centering

\includegraphics{<eps-file>}

\caption{<figure-caption>}\label{<figure-label>}

\end{figure}



**Fig. 1** This is a widefig. This is an example of long caption this is an example of long caption this is an example of long caption this is an example of long caption

**Table3**

Tableswhicharetoolongtofit,shouldbewrittenusingthe“sidewaystable”environmentasshownhere



Element1

1

Element

2



ProjectileEnergy

*σ*

*calc*

*σ*

*expt*

Energy

*σ*

*calc*

*σ*

*expt*



Element3990A11681547

±

12780A11661239

±

100

Element4500A961922

±

10900A12681092

±

40

Element5990A11681547

±

12780A11661239

±

100

Element6500A961922

±

10900A12681092

±

40



Note:Thisisanexampleoftablefootnotethisisanexampleoftablefootnotethisisanexampleoftablefootnotethisisanexampleoftablefootnote

thisisanexampleoftablefootnote.

1

Thisisanexampleoftablefootnote.

In case of double column layout, the above format puts figure captions/images to single column width. To get spanned images, we need to provide \begin{figure\*}

...\end{figure\*}.

For sample purpose, we have included the width of images in the optional argument of \includegraphics tag. Please ignore this.

# Algorithms, Program codes and Listings

Packages algorithm, algorithmicx and algpseudocode are used for setting algorithms in LATEX using the format:

\begin{algorithm}

\caption{<alg-caption>}\label{<alg-label>}

\begin{algorithmic}[1]

...

\end{algorithmic}

\end{algorithm}

You may refer above listed package documentations for more details before setting algorithm environment. For program codes, the “verbatim” package is required and the command to be used is \begin{verbatim}...\end{verbatim}.

Similarly, for listings, use the listings package. \begin{lstlisting} ... \end{lstlisting} is used to set environments similar to verbatim environment.

Refer to the lstlisting package documentation for more details.

A fast exponentiation procedure:

begin

for *i* := 1 to 10 step 1 do expt (2*,i* ) ;

newline () od Comments will be set flush to the right margin

where proc expt (*x,n*) ≡

*z* := 1;

do i f *n* = 0 then e x i t f i ;

do i f odd (*n*) then e x i t f i ;

comment : This is a comment statement;

*n* := *n/*2; *x* := *x* ∗ *x* od ; { *n >* 0 }; *n* :=1; *z* := *z* ∗ *x* od ; pr in t ( end



**Algorithm 1** Calculate *y* = *xn*

**Require:**



*n*

≥

0

∨

*x*

=0

**Ensure:** *y* = *xn*

1: *y* ⇐ 1

2: **if** *n <* 0 **then**

3: *X* ⇐ 1*/x* 4: *N* ⇐−*n*

5: **else** 6: *X* ⇐ *x* 7: *N* ⇐ *n*

8: **end if**

9: **while** **do**

10: **if** *N* is even **then**

11: *X* ⇐ *X* × *X*

12: *N* ⇐ *N/*2

13: **else**[*N* is odd]

14: *y* ⇐ *y* × *X*

15: *N* ⇐ *N* − 1

16: **end if**

17: **end while**



|  |
| --- |
| for i :=maxint to 0 do begin  { do nothing }  end ;  Write ( ’ Case i n s e n s i t i v e ’ ) ;  Write ( ’ Pascal keywords . ’ ) ; |

# Cross referencing

Environments such as figure, table, equation and align can have a label declared via the \label{#label} command. For figures and table environments use the \label{} command inside or just below the \caption{} command. You can then use the \ref{#label} command to cross-reference them. As an example, consider the label declared for Figure 1 which is \label{fig1}. To cross-reference it, use the command Figure\ref{fig1}, for which it comes up as “Figure 1”.

To reference line numbers in an algorithm, consider the label declared for the line number 2 of Algorithm 1 is \label{algln2}. To cross-reference it, use the command \ref{algln2} for which it comes up as line 2 of Algorithm 1.

## Details on reference citations

Standard LATEX permits only numerical citations. To support both numerical and author-year citations this template uses natbib LATEX package. For style guidance please refer to the template user manual.

Here is an example for \cite{...}: [1]. Another example for \citep{...}:[2]. For author-year citation mode, \cite{...} prints Jones et al. (1990) and \citep{...} prints (Jones et al., 1990).

All cited bib entries are printed at the end of this article: [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] and [13].

# Methods

Topical subheadings are allowed. Authors must ensure that their Methods section includes adequate experimental and characterization data necessary for others in the field to reproduce their work. Authors are encouraged to include RIIDs where appropriate.

**Ethical approval declarations** (only required where applicable) Any article reporting experiment/s carried out on (i) live vertebrate (or higher invertebrates), (ii) humans or (iii) human samples must include an unambiguous statement within the methods section that meets the following requirements:

1. Approval: a statement which confirms that all experimental protocols wereapproved by a named institutional and/or licensing committee. Please identify the approving body in the methods section
2. Accordance: a statement explicitly saying that the methods were carried out inaccordance with the relevant guidelines and regulations
3. Informed consent (for experiments involving humans or human tissue samples): include a statement confirming that informed consent was obtained from all participants and/or their legal guardian/s

If your manuscript includes potentially identifying patient/participant information, or if it describes human transplantation research, or if it reports results of a clinical trial then additional information will be required. Please visit ([https:// www.nature.com/nature-research/editorial-policies)](https://www.nature.com/nature-research/editorial-policies) for Nature Portfolio journals, [(https://www.springer.com/gp/authors-editors/journal-author/journal-authorhelpdesk/publishing-ethics/14214)](https://www.springer.com/gp/authors-editors/journal-author/journal-author-helpdesk/publishing-ethics/14214) for Springer Nature journals, or ([https://www. biomedcentral.com/getpublished/editorial-policies#ethics+and+consent)](https://www.biomedcentral.com/getpublished/editorial-policies#ethics+and+consent) for BMC.

# Discussion

Discussions should be brief and focused. In some disciplines use of Discussion or ‘Conclusion’ is interchangeable. It is not mandatory to use both. Some journals prefer a section ‘Results and Discussion’ followed by a section ‘Conclusion’. Please refer to Journal-level guidance for any specific requirements.

# Conclusion

Conclusions may be used to restate your hypothesis or research question, restate your major findings, explain the relevance and the added value of your work, highlight any limitations of your study, describe future directions for research and recommendations.

In some disciplines use of Discussion or ’Conclusion’ is interchangeable. It is not mandatory to use both. Please refer to Journal-level guidance for any specific requirements.

**Supplementary information.** If your article has accompanying supplementary file/s please state so here.

Authors reporting data from electrophoretic gels and blots should supply the full unprocessed scans for key as part of their Supplementary information. This may be requested by the editorial team/s if it is missing.

Please refer to Journal-level guidance for any specific requirements.

**Acknowledgments.** Acknowledgments are not compulsory. Where included they should be brief. Grant or contribution numbers may be acknowledged.

Please refer to Journal-level guidance for any specific requirements.

# Declarations

Some journals require declarations to be submitted in a standardised format. Please check the Instructions for Authors of the journal to which you are submitting to see if you need to complete this section. If yes, your manuscript must contain the following sections under the heading ‘Declarations’:

* Funding
* Conflict of interest/Competing interests (check journal-specific guidelines for which heading to use)
* Ethics approval
* Consent to participate
* Consent for publication
* Availability of data and materials
* Code availability
* Authors’ contributions

If any of the sections are not relevant to your manuscript, please include the heading and write ‘Not applicable’ for that section.

Editorial Policies for:

Springer journals and proceedings: <https://www.springer.com/gp/editorial-policies>

Nature Portfolio journals: <https://www.nature.com/nature-research/editorial-policies>

*Scientific Reports*: <https://www.nature.com/srep/journal-policies/editorial-policies>

BMC journals: <https://www.biomedcentral.com/getpublished/editorial-policies>

# Appendix A Section title of first appendix

An appendix contains supplementary information that is not an essential part of the text itself but which may be helpful in providing a more comprehensive understanding of the research problem or it is information that is too cumbersome to be included in the body of the paper.

# References

1. Campbell, S.L., Gear, C.W.: The index of general nonlinear DAES. Numer. Math. **72**(2), 173–196 (1995)
2. Slifka, M.K., Whitton, J.L.: Clinical implications of dysregulated cytokine production. J. Mol. Med. **78**, 74–80 (2000) <https://doi.org/10.1007/s001090000086>