Using Quantum-enhanced Support Vector Machines for Stellar Classification

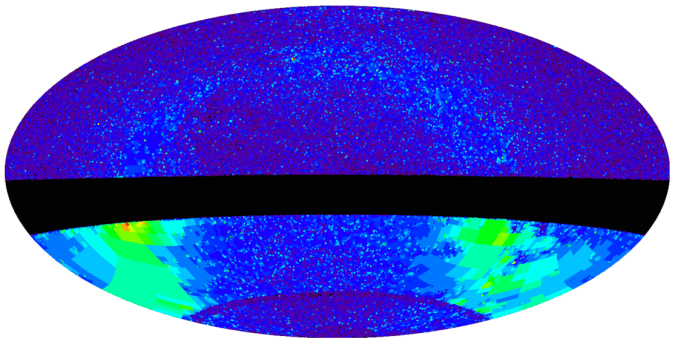
Start Date: 23/02/2022

Team Name:

Durchmusterung

Abstract:

This project aims to investigate the feasibility of employing quantum-enhanced support vector machines (QSVMs) for stellar classification based on spectral data from the open source dataset [1]. The core of the study involves designing and implementing a novel QSVM algorithm, which will be compared to traditional SVMs and Morgan–Keenan (MK) classification system, that is conventionally used in the field of stellar classification. Furthermore, the project seeks to explore the potential GPU acceleration techniques for the task of QSVM model training. Ultimately, this study will attempt to demonstrate the potential of quantum computing in enhancing the precision and efficiency of machine-learning approaches in astronomy.



1 Introduction

Stellar classification, the process of categorizing stars based on their spectral characteristics, is a fundamental aspect of astronomy. Accurately classifying stars provides valuable insights into their physical properties and evolutionary processes, aiding our understanding of the universe. Machine learning techniques, such as support vector machines (SVMs), have proven to be effective in classifying stars based on their spectral data [2]. However, the increasing complexity and size of astronomical datasets pose significant challenges to the performance and scalability of traditional SVMs.

Quantum computing has emerged as a promising approach to addressing these challenges. Quantum-enhanced SVMs have been shown to offer potential advantages in terms of improved accuracy and faster computation time for certain classification tasks. [3] Consequently, there is growing interest in exploring the feasibility of using quantum computing techniques for stellar classification.

This research project aims to investigate the potential of quantum-enhanced SVMs for stellar classification based on spectral data. By designing and implementing a novel quantum-enhanced SVM algorithm and comparing its performance to traditional SVMs, this study seeks to demonstrate the potential benefits of leveraging quantum computing methodologies in astronomy. The findings of this research can contribute to advancing the state-of-the-art in machine learning techniques for stellar classification and may have broader implications for other applications in astrophysics and beyond.

Types of Stars and Current Methods of Classification

Stellar classification is based on the properties of a star's spectrum, which can reveal information about its temperature, luminosity, and chemical composition [4]. The most widely used classification system is the Harvard spectral classification, which divides stars into seven categories (O, B, A, F, G, K, M) based on the strengths of their spectral lines. This system is based on the work of Annie Jump Cannon, who developed the original classification scheme in the early 20th century [5].

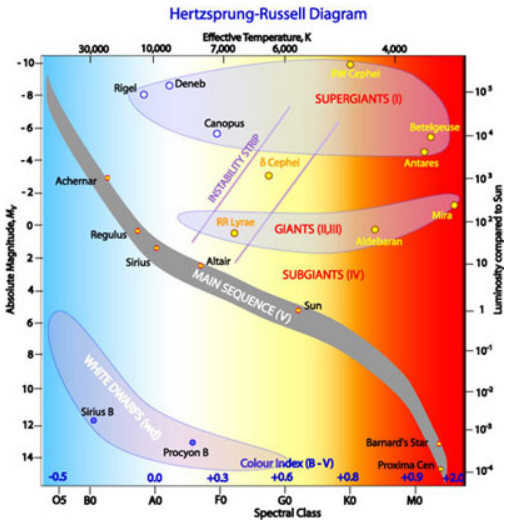


Figure 1: The Hertzsprung-Russell diagram the various stages of stellar evolution.

Each spectral type corresponds to a range of temperatures and luminosities, with O-type stars being the hottest and most luminous, and M-type stars being the coolest and least luminous. Within each spectral type, stars can be further divided based on their luminosity class, which is related to their size and evolutionary stage. The luminosity classes are denoted by Roman numerals I (supergiants), II (bright giants), III (giants), IV (subgiants), and V (main sequence stars) [6]. The Hertzsprung-Russell Diagram (HRD) is a plot of the luminosity of stars versus their surface temperature, and is a fundamental tool for studying stellar evolution [7]. This diagram reveals distinct regions where stars of different masses, sizes, and ages are located, and is essential for understanding the physical properties and behavior of stars. Typically, stars are classified on the HRD based on their spectral types and luminosities, with main sequence stars occupying a

diagonal band across the diagram and giants and supergiants occupying regions above and to the right of the main sequence. An example of an HRD is shown in Figure 1, where the stars are color-coded based on their spectral types, with O stars being the hottest and bluest, and M stars being the coolest and reddest.

The current methods of classifying stars are primarily based on human analysis of their spectra, which can be time-consuming and subject to variability [8]. Machine learning techniques, particularly SVMs, have been shown to be effective in automating the classification process and reducing human error [9]. However, as mentioned earlier, the performance and scalability of traditional SVMs are limited by the complexity and size of astronomical datasets. Hence, there is a need to explore novel approaches such as quantum-enhanced SVMs to improve the accuracy and efficiency of stellar classification [3].

2 Methodology

2.1 Quantum-enhanced Support Vector Machines (QSVM)

Machine learning is a powerful tool for solving complex problems in fields ranging from computer vision to natural language processing. One popular machine learning algorithm is the support vector machine (SVM), which is a powerful technique for classifying data. Recently, quantum computing has been explored as a potential tool for machine learning, and quantum support vector machines (QSVM) have been proposed as a quantum algorithm for binary classification.

In classical SVM, the algorithm takes a set of input data points and their corresponding labels, and finds the optimal hyperplane that separates the data into two classes. [10] The goal of the SVM algorithm is to find the hyperplane that maximizes the margin between the two classes. This hyperplane can be used to classify new data points based on which side of the hyperplane they lie, shown in Fig. 2. [11]

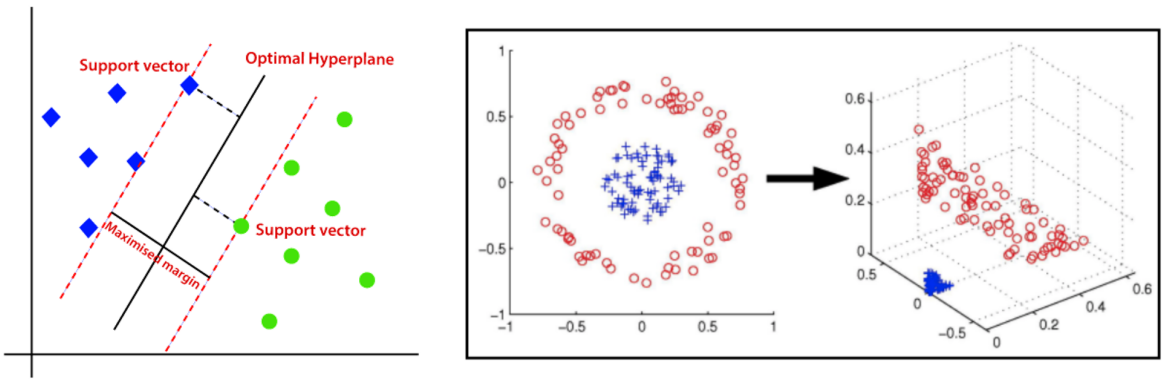


Figure 2: The concept of support vector machine [11]

QSVM is a quantum version of the classical support vector machine (SVM) algorithm, which is used for binary classification tasks [3]. In SVM, a hyperplane is used to separate the data points into two classes. The goal of SVM is to find the hyperplane that maximizes the margin between the two classes. The hyperplane is defined by a set of parameters that are learned during the training process. In QSVM, the classification is based on a quantum kernel function, which maps the input data points to a high-dimensional quantum state. The quantum state is then used to compute the inner product between two input data points. The inner product is then used to calculate the probability that the two data points belong to the same class. This probability is then used to classify the new data points.

The quantum kernel function used in QSVM is typically a variation of the classical kernel function used in SVM. One example is the quantum Gaussian kernel, which is defined as follows: [12]

$$K(x_i, x_j) = exp[\frac{-||x_i - x_j||^2}{2\sigma^2}] \tag{1}$$

where x_i and x_j are the input data points, and sigma is a hyperparameter that controls the width of

the kernel. The Gaussian kernel is commonly used in classical SVM, and the quantum version is computed using a quantum circuit.

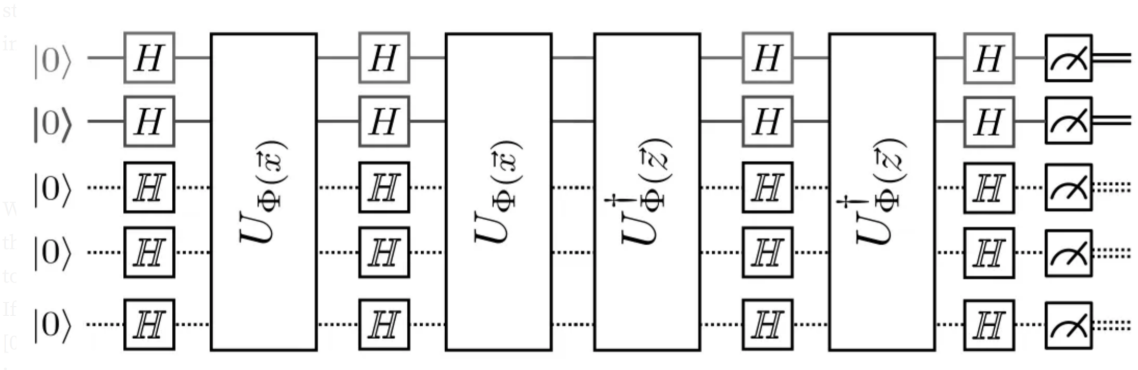


Figure 3: Quantum circuit os QSMV. The estimated overlap between two quantum states for a circuit depth of S=2. [12]

One of the main advantages of QSVM over classical SVM is its ability to take advantage of quantum parallelism. In classical SVM, the kernel function is typically computationally expensive, and the SVM algorithm can be slow for large datasets. QSVM can use quantum parallelism to simultaneously evaluate the quantum kernel function for multiple data points, which can speed up the computation significantly. For example, if we have n data points, we can evaluate the kernel function for all n^2 pairs of data points simultaneously using a quantum circuit. This can be done in O(log n) time on a quantum computer, compared to $O(n^2)$ time on a classical computer.

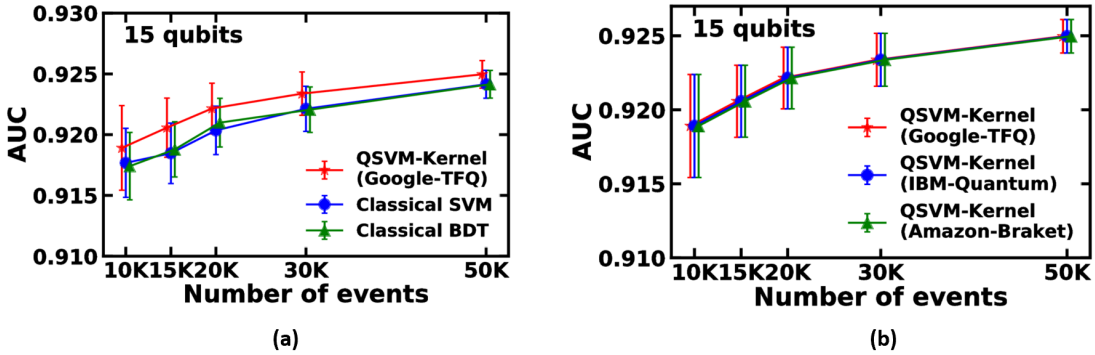


Figure 4: Comparison of classifiers for the *ttH* analysis dataset, showing the AUC as a function of dataset size. The results indicate that QSVM-Kernel outperforms classical SVM and BDT, and performs well on multiple quantum simulators [13]

In Fig. 4, the CERN's paper [13] compared the performance of various classifiers, including QSVM-Kernel, classical SVM, and classical BDT, for the *ttH* analysis dataset, using the AUC as a metric. The results showed that QSVM-Kernel outperformed classical SVM and BDT, especially for larger datasets. In addition, the study evaluated the performance of the QSVM-Kernel on multiple quantum simulators, demonstrating its versatility and potential for use in solving problems in high-energy physics and astrophysics. [13] Given its high accuracy and ability to perform well on large and complex datasets, QSVM-Kernel may be a valuable tool for researchers in these fields.

In addition to quantum parallelism, Q-SVM can also exploit the power of quantum entanglement to enhance the classification performance. Entanglement is a fundamental property of quantum systems, which allows two or more particles to be correlated in a way that is not possible in classical systems. By exploiting entanglement, QSVM can potentially achieve better classification performance than classical. Q-SVM can take advantage of quantum parallelism and quantum entanglement to enhance the classification performance. These quantum advantages can potentially speed up the computation and improve the classification accuracy, especially for large and complex datasets - like the case we want to solve in this work.

2.2 Star Dataset for QSVM Classification

The Star Categorization - Giants and Dwarfs dataset on Kaggle is a collection of stellar parameters for a sample of stars in our galaxy. [1] The dataset includes over 100,000 stars and provides information on the stars' luminosity, temperature, radius, and mass.

The Star Categorization - Giants and Dwarfs dataset provides information on over 100,000 stars in our galaxy and is categorized into two groups: giants and dwarfs. Giants are stars that are more luminous and larger than the sun, while dwarfs are stars that are less luminous and smaller than the sun. This categorization is based on the stars' luminosity and radius, which are estimated from their spectral data. The dataset was compiled by researchers at the Vines Center for Astronomy at Michigan State University and was originally used to study the relationship between stellar parameters and the occurrence of exoplanets. The dataset is publicly available on Kaggle and can be used for various research purposes, such as studying stellar evolution, exoplanet detection, and machine learning applications in astronomy. By analyzing the dataset, researchers can gain insights into the properties and behavior of stars, and develop machine learning models for classifying stars based on their stellar parameters. These models can be used to identify new exoplanets and to study the evolution of stars over time.

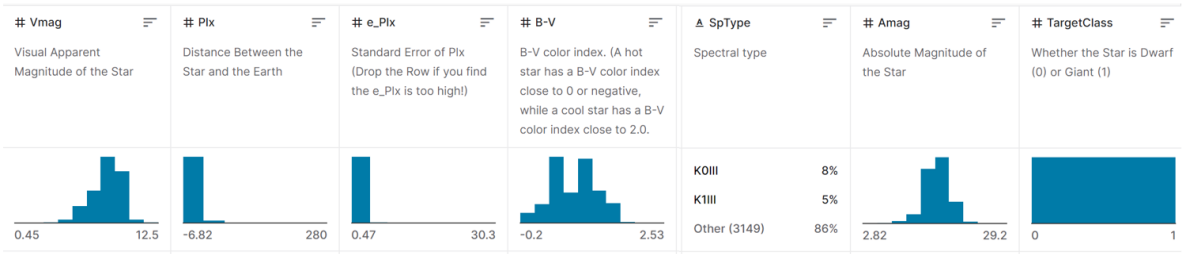


Figure 5: The statistics of star dataset

In this work, we will use this dataset to explore various questions about the properties and behavior of stars, and to develop machine learning models for classifying stars as giants or dwarfs based on their stellar parameters. The dataset can also be used as a benchmark for testing the performance of quantum machine learning algorithms in astronomy and astrophysics.

2.2.1 Data Pre-Processing for Astrophysical Data

Data preprocessing is an essential step in machine learning, as it helps to prepare the data for analysis and improve the performance of the machine learning models. In the case of the Star Categorization - Giants and Dwarfs dataset, the data preprocessing involves several steps to clean and transform the data.

The first step in data preprocessing is to remove any duplicate entries or missing values in the dataset. This ensures that the dataset is complete and consistent, and that there are no errors or inconsistencies in the data.

Next, the data is standardized to remove any bias or scale differences that may be present in the dataset. Standardization is a crucial step in data preprocessing that involves scaling the features of a dataset to remove any bias or scale differences that may be present. Standardization is performed by subtracting the mean of each feature from the data and dividing it by the standard deviation. This results in a feature that has a mean of 0 and a standard deviation of 1, which makes it easier to compare different features with each other. Standardization ensures that all features have equal importance in the analysis, and that the machine learning models are not influenced by differences in the scale of the features. This step is particularly important in machine learning, where the scale of the features can significantly impact the accuracy and performance of the models. Without standardization, the machine learning models may be biased towards certain features and may not perform well on new data. By standardizing the data, the machine learning models can more accurately identify patterns and relationships between the features, leading to better performance and more reliable predictions.

After standardization, the data is split into training and testing sets to evaluate the performance of the machine learning models. The training set is used to train the models, while the testing set is used to evaluate the accuracy and generalization of the models.

Finally, the data is encoded to prepare it for machine learning models. This involves transforming the categorical data into numerical data, which can be used in machine learning algorithms. One common technique for encoding categorical data is one-hot encoding, which creates binary columns for each category in the dataset.

	Vmag	Plx	e_Plx	B-V	SpType	Amag	TargetClass
0	10.00	31.660000	6.19	1.213	K7V	22.502556	1
1	8.26	3.210000	1.00	1.130	K0III	15.792525	0
2	8.27	12.750000	1.06	0.596	F9V	18.797552	1
3	6.54	5.230000	0.76	1.189	K1III	15.132508	0
4	8.52	0.960000	0.72	0.173	B8V	13.431356	1
...	...	...	...	...	...	...	...
39547	5.83	0.170000	0.52	0.474	B7Iab	6.982244	0
39548	7.05	18.120001	0.92	0.424	F5V	18.340790	1
39549	9.21	3.890000	1.46	0.227	A1IV	17.159748	1
39550	9.01	2.130000	1.46	1.467	M5III	15.651898	0
39551	9.12	3.820000	0.79	0.480	F5V	17.030317	1

39552 rows × 7 columns

Figure 6: Dataset after pre-processing remains 39552 data points with 6 parameters and 1 label.

2.3 Compute Resource Estimation

2.3.1 Amazon Web Services

Amazon Web Services (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to businesses, individuals, and governments [14]. AWS offers a wide range of services, including computing power, storage, and databases, that enable businesses to scale and grow their operations without having to invest in expensive hardware and infrastructure. Among the services that AWS provides, Amazon Bracket is a fully-managed service that provides access to quantum computing technology from multiple providers in a single place. With Amazon Bracket, you can design your own quantum algorithms, test and troubleshoot them on simulated quantum computers running on Amazon EC2, and run them on your choice of different quantum hardware providers.

2.3.2 NVidia - cuQuantum

The NVidia cuQuantum SDK is a toolkit that allows researchers to simulate quantum circuits using GPUs [15]. GPUs are specialized hardware that are designed to perform parallel computations efficiently. This makes them well-suited for accelerating the computation-intensive tasks involved in quantum computing, such as simulating quantum circuits.

By using the cuQuantum SDK on a GPU, you can take advantage of its parallel processing capabilities to accelerate our quantum computing calculations. This can reduce the time required to simulate quantum circuits and optimize quantum algorithms with a very huge dataset - like our star dataset, which can be very time-consuming on classical computers. [16]

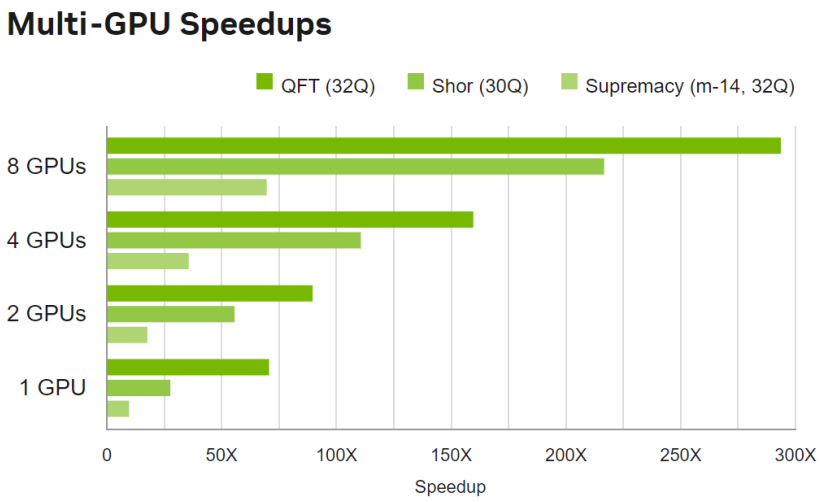


Figure 7: GPU speedup of cuQuantum [15]

Using a GPU with the NVidia cuQuantum SDK can not only result in faster computation times but also enable researchers to simulate larger and more complex quantum circuits. GPUs have significant amounts of memory and processing power, which can be utilized to simulate circuits with thousands or even millions of qubits. This capability can enable researchers to explore new quantum computing algorithms and applications that were previously infeasible to simulate on classical computers. In our work, we anticipate that training with over 30,000 samples will be time-consuming. By utilizing cuQuantum, we can efficiently achieve our results within one week and quantitatively benchmark the performance between CPU and GPU. This work will undoubtedly be beneficial to researchers in the fields of high-energy physics and astrophysics.

3 Future Directions

QSVM is a quantum machine learning algorithm that has potential applications in various fields in the future. In particular, it could be used in physics for tasks such as the classification of data, anomaly detection, and optimization problems. For example, QSVM could be used in particle physics for classifying data from particle detectors, like Higgs boson [17]. The detection of the Higgs boson at the Large Hadron Collider involved the analysis of massive amounts of data, which was processed using classical machine-learning techniques. QSVMs could potentially be used to improve the accuracy and efficiency of this data analysis, by leveraging the quantum properties of a quantum computer to perform calculations that would be difficult or impossible to perform on a classical computer.

Another potential of QSVM is in the string theory. String theory is a theoretical framework that attempts to unify all known fundamental forces of nature, including gravity, into a single theory [18]. One of the challenges of string theory is that it predicts the existence of many more dimensions than the three spatial dimensions and one time dimension that we experience in our everyday lives. This makes calculations in string theory extremely complex, and it is difficult to perform simulations or make predictions based on the theory. Potentially, QSVMs can simulate the behavior of string-like objects in higher-dimensional space.

Overall, the future of QSVMs is bright, and they have the potential to revolutionize many fields by providing faster and more accurate classification and prediction of complex data sets. As the development of quantum computing technology advances, we can expect to see more and more applications of QSVMs in a wide range of industries and research fields.

References

[1] W.-F. Ku, *Star categorization giants and dwarfs dataset, vinesmsuic.* https://www.kaggle.com/datasets/vinesmsuic/star-categorization-giants-and-dwarfs, Jul 2020.

[2] C. A. L. Bailer-Jones and M. Irwin, *Automated classification of hipparcos periodic variable stars using support vector machines*, Astronomy & Astrophysics **385** (2002), pp. 95–105.

[3] P. Rebentrost, M. Mohseni and S. Lloyd, *Quantum support vector machine for big data classification*, Physical Review Letters **113** (2014), p. 130503.

[4] D. F. Gray, *The observation and analysis of stellar photospheres*, Cambridge University Press, 2021.

[5] P. Keenan and E. Kellman, *An atlas of stellar spectra. with an outline of spectral classification. astrophys*, Monographs. Chicago (1943).

[6] A. N. Cox, *Allen's astrophysical quantities*, Springer, 2015.

[7] E. Hertzsprung and H. N. Russell, *On the relation between the spectra and other characteristics of the stars.*, Astrophysical Journal **67** (1928), pp. 40–79. doi:10.1086/143167.

[8] J. H. Lee and K. M. Seo, *A spectral classification of stars using decision tree and ensemble learning*, Journal of the Korean Astronomical Society **50** (2017), pp. 173–186.

[9] C. A. L. Bailer-Jones, R. Andrae, B. Arcay and et al., *Estimating distances from parallaxes. iv. distances to (some) open clusters using GAIA data*, Astronomy & Astrophysics **618** (2018), p. A93.

[10] C. Cortes and V. Vapnik, *Support-vector networks*, Machine learning **20** (1995), pp. 273–297.

[11] W. S. Noble, *What is a support vector machine?*, Nature biotechnology **24** (2006), pp. 1565–1567.

[12] V. Havlíček et al., *Supervised learning with quantum-enhanced feature spaces*, Nature **567** (2019), pp. 209–212.

[13] S. L. Wu et al., *Application of quantum machine learning using the quantum kernel algorithm on high energy physics analysis at the lhc*, Physical Review Research **3** (2021), p. 033221.

[14] Amazon Web Services, Inc., *Amazon Braket: Get started with quantum computing.* https://aws.amazon.com/braket/, 2021. Accessed: February 23, 2023.

[15] NVIDIA, *cuQuantum SDK: Simulating quantum circuits on GPUs.* https://developer.nvidia.com/cuquantum-sdk, 2021. Accessed: February 23, 2023.

[16] S. Stanwyck, H. Bayraktar and T. Costa, *cuquantum: Accelerating quantum circuit simulation on gpus*, in APS March Meeting Abstracts, vol. 2022, 2022, pp. Q36–002.

[17] G. Aad et al., *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, Phys. Lett. B **716** (2012), pp. 1–29. arXiv:1207.7214, doi:10.1016/j.physletb.2012.08.020.

[18] J. Polchinski, *String theory. Vol. 1: An introduction to the bosonic string*, Cambridge monographs on mathematical physics, Cambridge University Press, 1998. doi:10.1017/CBO9780511816079.