Google Summer of Code Proposal

Quantum Variational Autoencoders for HEP Analysis at the LHC

Tom Magorsch*

TU Dortmund University, Department of Physics,

+49 (157) 57285360, GitHub

In the search for physics beyond the standard model physics, the growing amount of data and the evasive of the signals that are being searched for, calls for new methods. A prominent method is the use of autoencoders to search for anomalous events in an unsupervised manner. In project I implement a quantum autoencoder to perform anomaly detection on jet images. The goal is to distinguish QCD jets from anomalous jets with a different production mechanism. I train the quantum autoencoder on standard model data and compare it's performance as an anomaly tagger to a classical autoencoder, to study the potential use of quantum machine learning methods to enhance HEP analyses.

Contents

*Electronic address: tom.magorsch@tu-dortmund.de

I. RESEARCH CONTEXT

With the growing amount of data the LHC produces, and a potential HL-LHC will produce, sophisticated machine learning methods are an important research direction in order to disentangle the evasive beyond standard model (BSM) physics from standard model (SM) signals. Apart from classical machine learning techniques, a growing research interest is directed towards the application of state of the art deep learning methods to typical HEP challenges [1, 2] and even enable new search strategies. With the progress of quantum computing in the NISQ era, the question arises if the new technology can be used to enhance BSM searches, potentially opening new possibilities, to discover yet unseen signals in the data.

A. New physics search with anomaly detection techniques

Due to the apparently evasive nature of new physics (NP) a large body of potential NP models has been developed. An effective search paradigm to deal with this are model independent strategies, which do not assume a specific NP phenomena to search for, but rather explore any phenomena that deviates from the SM. One possible model independent analysis is anomaly detection. A machine learning model trained on SM data thereby tries to uncover events that are anomalous and can therefore be events of some unknown BSM process. Anomaly detection is an unsupervised learning framework and can therefore be used for model independent searches, e.g. [3, 4]. One application of NP searches using anomaly detection are QCD jets [4–6]. Here the goal is to distinguish QCD jets from jets with a different production mechanism. A popular method for deep anomaly detection is the use of autoencoders.

B. Related work

Recently the separation of QCD jets and top- and W-jets was reported in Ref. [4, 5]. In both works, the authors use the dataset [7] for the QCD jets and [8] for the anomalous jets. The dataset for anomalous jets contains e.g. jets from W' with different masses. The datasets contain the four vectors of the constituents of the jets. Both papers [4, 5] compute images from the constituents p_T following [9]. In [4] the authors use a classical convolutional architecture and explore different metrics to tag the anomalies.

Another common benchmark task is the tagging of top jets [6, 10], with a similar rationale. The respective dataset [11] contains top- and QCD jets. Again the tagging of the top jets is used as benchmark, but the methods can also be applied to the search for unknown BSM signatures.

Furthermore [14] is a dataset of already pixelated 25×25 QCD and W-boson jet images introduced in [15]. This dataset is equally suitable for an anomaly detection analysis where e.g. an autoencoder could learn the structure of QCD jets to spot the anomalous W-boson jets.

A study concerning the use of a quantum autoencoder (QAE) for unsupervised new physics

search has been reported in [12]. The authors show that the quantum autoencoder outperforms a classical autoencoder (CAE) in spotting a simulated heavy higgs. The autoencoder is trained on standard model data using gradient descent, where the loss function is based on the reconstruction fidelity which is calculated by a SWAP test [13]. The reconstruction fidelity is then used as a discriminant to label anomalous events.

II. RESEARCH GOALS

This project aims to demonstrate the application of quantum machine learning (QML) methods to HEP analysis tasks and explore if and how it can improve classical analyses. An important part of the project is therefore the comparison between QML and classical algorithms. Furthermore different architectures of QML models and hybrid quantum-classical can be explored and compared. Such a proof of concept implementation can lay a foundation for new QML algorithms to build upon.

As goals for the GSoC project I would like to propose the following:

- A reusable implementation of a quantum autoencoder, see Sec. III B for details.
- Training the autoencoder on jet data in conjunction with a dimensionality reduction technique to find anomalous jets
- Training a comparable CAE on the jet data
- Compare the quantum and CAE based on number of parameters and convergence in training
- Summarize the findings and results in a short paper

Depending on the progress of these initial goals, I would also like to explore a hybrid model. The quantum autoenencoder can therefore optionally be replaced by a hybrid model, see Sec. III C. Ideally, I would like to implement both architectures and compare the performance, but I will make this dependent on the mentors feedback and the progress on the first implementation of the full quantum autoencoder. All implementations will be performed in python using Cirq and TF-Quantum.

III. METHODOLOGY

In this section I will outline my proposed Methodology. For parts of the project I will suggest different possibilities which I would like to try out to see which seems the most promising and also hope to receive feedback from the mentors.

A. Dataset

As mentioned in Sec. I B I find three publicly available datasets to be suitable:

[14] 873k QCD and W-boson 25×25 jet images

[7, 8] 706k QCD jets and 200k jets for different anomalous signatures respectively, each containing the four vectors of the jets constituents

[11] 2M QCD and top jets, each containing the four vectors of the jets constituents

The approach for all three datasets would be very similar. The SM QCD jets are used for training, while the other jets serve as signal in validation and test data. The signal is supposed to be uncovered in an unsupervised way, without including it in the training, by tagging it according to the reconstruction error of the autoencoder. The latter two datasets contain the four momenta of the jet constituents, which would be converted to images following [9].

For any of the three datasets, I would propose to split the QCD jets into training-/validation-/testing-sets with 80%/10%/10%. The anomalous jets can be split 50%/50% between validation and test data. Although the optimization of hyperparameters will only be performed on the reconstruction efficiency of the autoencoder, and therefore without the use of anomalous data, it is necessary to reserve anomalous samples for validation in order to determine a threshold for anomaly tagging.

B. Quantum architecture

A possible model architecture is shown in Fig. 1. Here the model generally consists of two steps:

1. A classical ML technique to reduce the dimension of the data

2. The Quantum Auto Encoder working on the reduced data

The first step is necessary as e.g. 25×25 images would be too large for a quantum circuit to train on. Generally there are different possibilities to reduce the dimension of the data. A simple choice would e.g. be a PCA or SVD on the raw jet images. After the dimensionality reduction the data needs to be prepared for encoding into a quantum state. There are different encoding methods, with the most simple one being the angle encoding [16] which converts a classical data-point x_i to a quantum state $\cos(x_i)\,|0\rangle + \sin(x_i)\,|1\rangle$. To minimize the number of qbits, used, the amplitude encoding could be implemented, which encodes the data in the amplitude of the n-qbit hilbertspace, see Ref. [16].

The actual quantum autoencoder architecture is shown in Fig. 1 taken from [17, 18]. For the sake of simplicity the example assumes three dimensional data in the angle encoding. In this case the encoding block of the autoencoder, which consists of an arbitrary number of layers with trainable parameters, acts on the first three qbits. In principle different structures for the encoder and decoder circuit should be possible [17–19], but I would propose the one shown in Fig. 2 as it was reported in [21] to be the most effective one. However at this point it might be worth trying different possible circuits and choose the best performing one. The decoder circuit has the same

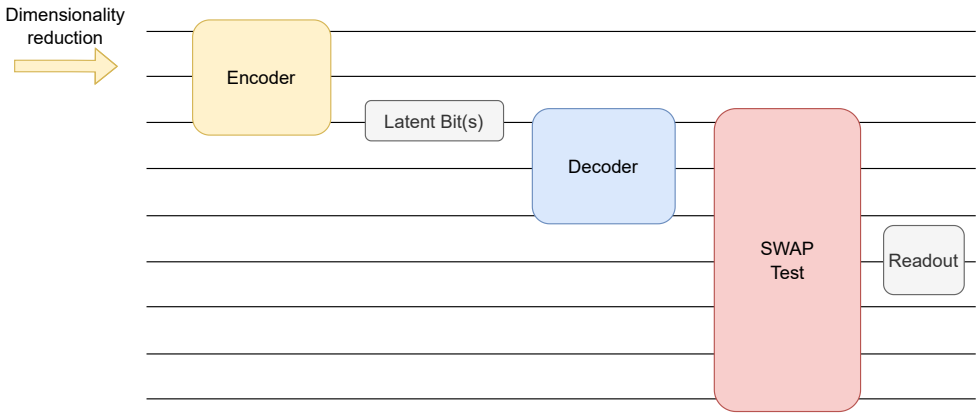
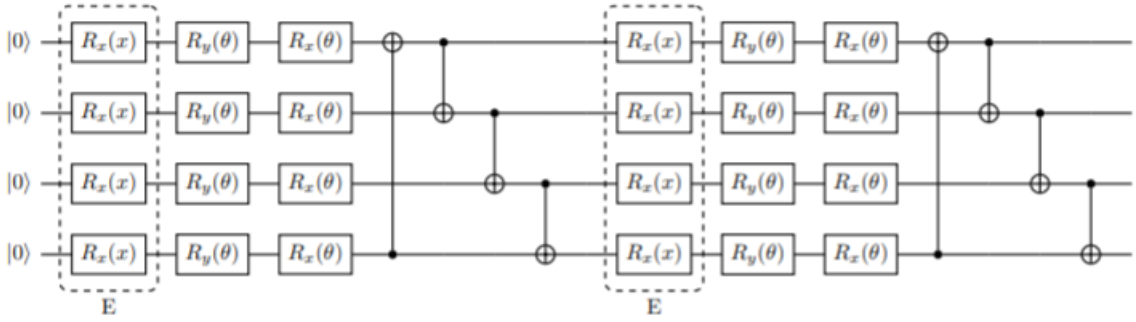


FIG. 1: The model architecture consisting of some method to reduce the dimension of the data which is then fed into the quantum encoder. In the shown example the input has three qbits and one latent qbit.



Circuit 10

FIG. 2: Proposed circuit for the encoder and decoder network of the QAE in the case of four data qbits. Figure taken from [21]

structure like the encoder. However it only overlaps with the data qbits, the encoder acted on, at the latent qbits, as shown in Fig. 1. The rest of the Decoder acts on trash qbits. After the Decoder, a SWAP test is performed as introduced in [13], see also the explanation in [20]. The result of the SWAP test is read out at the end, and equals the fidelity of the output of the decoder and the input data. It can therefore be used to compute the loss.

For the QAE architecture described above $3 \times n - l + 1$ qbits are needed, where n are the number of data qbits and l the number of latent qbits. The number of qbits n and l are hyperparameters that need to be optimized.

C. Hybrid architecture

I am interested in exploring hybrid models, to combine the strengths of both classical and quantum machine learning to a practical model. Namely, the classical neural network can compress

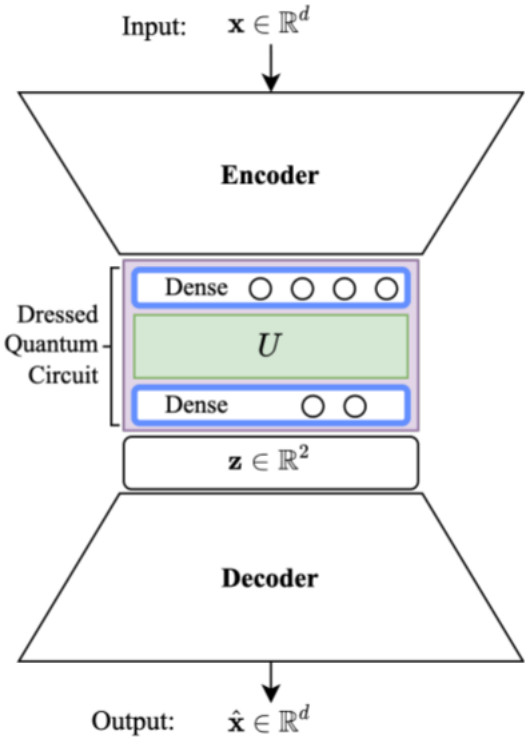


FIG. 3: Hybrid QAE composed of a classical encoder and decoder connected by a variational quantum circuit U. Figure taken from [22].

the classical data and learn important features e.g. with a CNN architecture. For this project I propose an architecture similar to the one reported in Ref. [22]. In their work, the authors include a "dressed" quantum circuit between a classical encoder and decoder as shown in Fig. 3. The dressed quantum circuit is composed of a quantum circuit with trainable parameters between two classical layers with n and l qbits. With this architecture the SWAP test is not necessary. Instead a simple MSE loss can be used for the input data x_i and the reconstructed samples $\hat{x}_i$. For the classical architecture I propose multiple blocks of convolutional- max-pooling- and batch-normalization-layers respectively. The number of filters and filter size are hyperparameters that will be optimized on the validation dataset. Like in the case of the quantum autoencoder, the training of this hybrid autoencoder (HAE) can be carried out in tensorflow with a standard optimizer like Adam or RMSprop.

D. Evaluation

For a given dataset a model should be evaluated on it's ability to find anomalies in an unseen test dataset.

In general hyperparameters can be optimized based on the reconstruction efficiency of the QCD jets in the validation dataset. For the pure QAE the reconstruction efficiency is measured by the

fidelity calculated from the SWAP test. For the HAE the MSE can be used as reconstruction efficiency.

Since autoencoders perform anomaly tagging based on the reconstruction error, a threshold needs to be chosen, when a given sample is deemed anomalous. To do so I will first plot the reconstruction efficiency of the anomaly tagger in terms of the chosen anomaly threshold using the validation dataset. A suitable threshold can then be chosen based on the results.

The final evaluation is performed on the test set when all hyperparameters are fixed. To estimate the performance of the models we can report the roc curve and auc. For a meaningful result of the project it is important to show if the quantum models can outperform the classical counterpart by comparing the roc curves.

IV. PROJECT PLAN

As the project is advertised for 175 hours, I am planning on evenly distributing the work to 2-3 hours per day. Currently I have nothing special planned for the summer so I will have enough time to work on the project besides my usual research work.

In general, the project outlined in this draft is only a proposal for possible work. I would be happy to further discuss details with the mentors during the community bonding period. I am also open to e.g. use a different dataset or adjust the proposed models to mentors suggestions.

In the end of the project I will produce a short document summarizing the project and showing the results. If the results are interesting I would be happy to extend this document to a full publication in collaboration with my mentors. As the work on the project itself will already take up some time I would work on polishing the draft and bringing the project in a publication worthy state after the GSoC deadline.

During my work I will provide code and additional material in an open repository to make my progress transparent.

A. Timeline

In this section I give a detailed timeline with milestones. Possible adjustments during the development of the project are possible. I plan the project in sprints of two weeks, each with a defined increment, which I think is achievable in this time frame. My proposed timeline is outlined in Tab. IV A.

V. ABOUT ME

I recently graduated with a master in physics and will start my PhD next month at the University of Dortmund. I work in particle phenomenology and am interested in new methods to discover BSM physics. In addition, I find machine learning very intriguing and like to explore new ML techniques.

Milestone	Due date	Description
Bonding period	20.05.	
Establish project details	03.06.	Meet with the mentors to agree on project details including project goals, dataset, model architecture and timeline
Setup	10.06.	Setup a public repository for code and notes and obtain the dataset. Write utility functions to load the data and to create train/val/test-split.
Official Coding begins	13.06.	
Data reprocessing	24.06.	Clean dataset, create images from jet data in case of jet constituent dataset. Apply PCA on jet images and write quantum encoding to prepare for data for full QAE.
Implement QAE	08.07.	Implement the QAE from Sec. III B and train it on the data.
Implement CAE	15.07.	Implement a CAE which trains on the same input data as the QAE.
Optimization & evaluation	29.07.	Optimize hyperparameters of the QAE and CAE on validation data. Determine the anomaly threshold for tagging and calculate roc curves and aucs.
Implement HAE	12.08.	Implement the HAE from Sec. III C and train it.
HAE evaluation	26.08.	Optimize hyperparameters and evaluate hybrid model with roc curve. A final plot should contain roc curve of the CAE, QAE and HAE.
Finish report paper	09.09.	Finish the final project report with a detailed project description and all results.
Finish of GSoC	12.09.	
Plan for further work	23.09.	Discuss project and future plans with mentors

TABLE I: The proposed timeline composed of two week sprint milestones.

I have many years of experience programming in python and it's ecosystem. Furthermore I worked with git, C++ and many other programming languages before. For more information please see my CV, which I e-mailed together with my evaluation tasks.

[1] M. Feickert and B. Nachman, [arXiv:2102.02770 [hep-ph]].

[2] M. Abdughani, J. Ren, L. Wu, J. M. Yang and J. Zhao, Commun. Theor. Phys. **71** (2019) no.8, 955 doi:10.1088/0253-6102/71/8/955 [arXiv:1905.06047 [hep-ph]].

[3] G. Kasieczka, B. Nachman, D. Shih, O. Amram, A. Andreassen, K. Benkendorfer, B. Bortolato, G. Brooijmans, F. Canelli and J. H. Collins, *et al.* Rept. Prog. Phys. **84** (2021) no.12, 124201 doi:10.1088/1361-6633/ac36b9 [arXiv:2101.08320 [hep-ph]].

[4] K. Fraser, S. Homiller, R. K. Mishra, B. Ostdiek and M. D. Schwartz, JHEP **03** (2022), 066 doi:10.1007/JHEP03(2022)066 [arXiv:2110.06948 [hep-ph]].

[5] T. Cheng, J. F. Arguin, J. Leissner-Martin, J. Pilette and T. Golling, [arXiv:2007.01850 [hep-ph]].

[6] T. Finke, M. Krämer, A. Morandini, A. Mück and I. Oleksiyuk, JHEP **06** (2021), 161 doi:10.1007/JHEP06(2021)161 [arXiv:2104.09051 [hep-ph]].

[7] Leissner-Martin, J., Cheng, T. & Arguin, J. QCD Jet Samples with Particle Flow Constituents. (Zenodo,2020,7), `https://doi.org/10.5281/zenodo.4641460`

[8] Cheng, T. Test sets for jet anomaly detection at the LHC. (Zenodo,2020,4), `https://doi.org/10.5281/zenodo.3774560`

[9] S. Macaluso and D. Shih, JHEP **10** (2018), 121 doi:10.1007/JHEP10(2018)121 [arXiv:1803.00107 [hep-ph]].

[10] T. Heimel, G. Kasieczka, T. Plehn and J. M. Thompson, SciPost Phys. **6** (2019) no.3, 030 doi:10.21468/SciPostPhys.6.3.030 [arXiv:1808.08979 [hep-ph]].

[11] Kasieczka, G., Plehn, T., Thompson, J. & Russel, M. Top Quark Tagging Reference Dataset. (Zenodo,2019,3), `https://doi.org/10.5281/zenodo.2603256`

[12] V. S. Ngairangbam, M. Spannowsky and M. Takeuchi, [arXiv:2112.04958 [hep-ph]].

[13] Buhrman, H., Cleve, R., Watrous, J. & Wolf, R. Quantum Fingerprinting. *Phys. Rev. Lett.*. **87**, 167902 (2001,9), `https://link.aps.org/doi/10.1103/PhysRevLett.87.167902`

[14] Nachman, B., Oliveira, L. & Paganini, M. Pythia Generated Jet Images for Location Aware Generative Adversarial Network Training. (Zenodo,2017,2), `https://doi.org/10.17632/4r4v785rgx.1`,

[15] L. de Oliveira, M. Paganini and B. Nachman, Comput. Softw. Big Sci. **1** (2017) no.1, 4 doi:10.1007/s41781-017-0004-6 [arXiv:1701.05927 [stat.ML]].

[16] Weigold, M., Barzen, J., Leymann, F. & Salm, M. Expanding Data Encoding Patterns For Quantum Algorithms. *2021 IEEE 18th International Conference On Software Architecture Companion (ICSA-C)*. pp. 95-101 (2021)

[17] Romero, J., Olson, J. & Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quantum Science And Technology*. **2**, 045001 (2017,8), `https://doi.org/10.1088%252F2058-9565%252Faa8072`

[18] Bravo-Prieto, C. Quantum autoencoders with enhanced data encoding. *Machine Learning: Science And Technology*. **2**, 035028 (2021,7), `https://doi.org/10.1088/2632-2153/ac0616`

[19] Farhi, E. & Neven, H. Classification with Quantum Neural Networks on Near Term Processors. (arXiv,2018), https://arxiv.org/abs/1802.06002

[20] Schuld, M., Sinayskiy, I. & Petruccione, F. An introduction to quantum machine learning. *Contemporary Physics*. **56**, 172-185 (2014,10), `https://doi.org/10.1080%252F00107514.2014.964942`

[21] A. Sakhnenko, C. O'Meara, K. J. B. Ghosh, C. B. Mendl, G. Cortiana and J. Bernabé-Moreno, [arXiv:2112.08869 [quant-ph]].

[22] Rivas, P., Zhao, L. & Orduz, J. Hybrid Quantum Variational Autoencoders for Representation Learning. *The 19th International Conference On Scientific Computing (CSC 2021).* (2021)