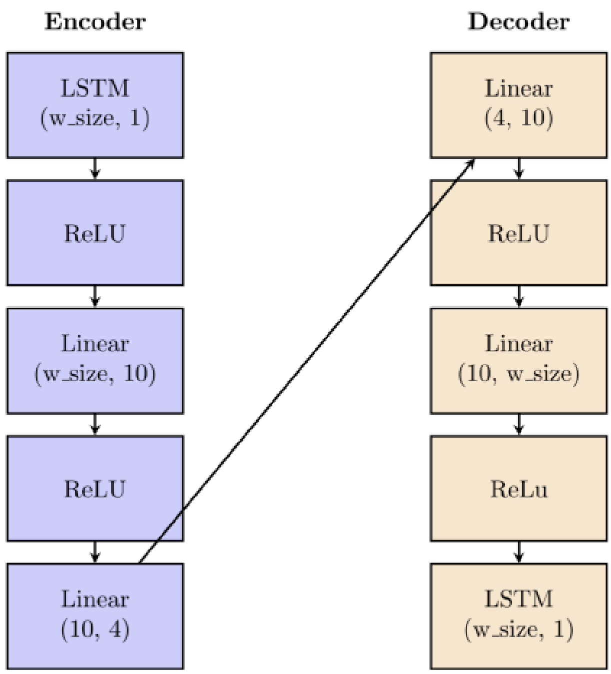
CPEN400Q Final Report

Applying Quantum Autoencoders for Time Series Anomaly Detection

Group 2: Buland, Lucas, Lisa, Laura

Background Theory

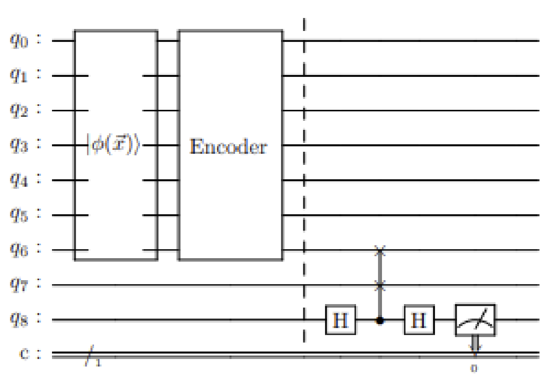
In this project, we explored using quantum autoencoders for anomaly detection.



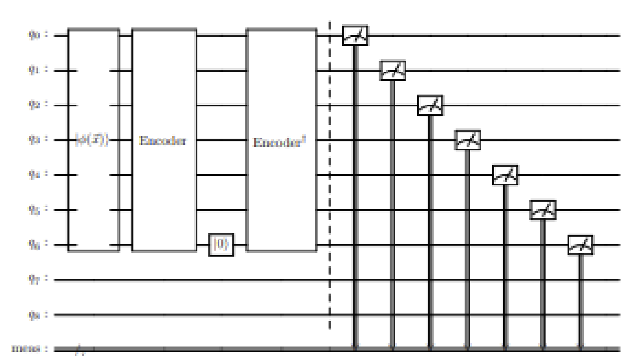
Autoencoders have been around for several decades [1]. Autoencoders are useful for anomaly detection in a very intuitive way. Essentially, they compress data through an encoder, retaining only the most important details (most relevant and frequently occurring features). The compressed data is then reconstructed with a decoder. The reconstruction causes parts with anomalies to get erased or become poorly reconstructed. Anomalies can then be detected by comparing the original input and the reconstructed output.

While classical autoencoders are well-studied. quantum autoencoders are still largely unexplored. In this project, we implemented both a MSE-based (mean squared error) and SWAP-based quantum encoder for anomaly detection.

In the implementation, there is a trained and trainable autoencoder. Essentially, the trained autoencoder is an MSE-based autoencoder that is more similar to a classical autoencoder which has an encoder and decoder. It detects anomalies with the method of reconstruction. The trainable autoencoder is the SWAP-based autoencoder. It uses an encoder, but it is not followed by a decoder. Instead, it is followed with a SWAP test to detect anomalies.



(a) Trainable quantum autoencoder architecture

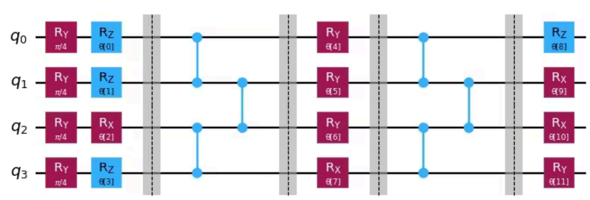


(b) Trained quantum autoencoder architecture

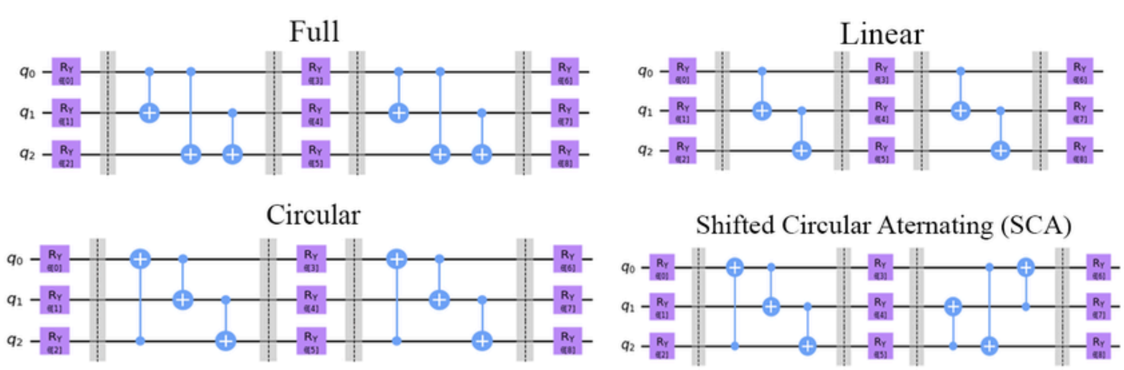
Trained quantum autoencoder
This autoencoder consists of an encoder and decoder. In our implementation, the encoder compresses the data of 7 qubits into 6 qubits. The 7th “trash” qubit is set to as close to 0 as possible, because it should contain no data. The further it is from 0, the more likely it is that there is an anomaly. The data is then passed through a decoder to spread the data across 7 qubits again. The qubits are then measured, and the input is compared with the output.

Trainable quantum autoencoder
This autoencoder consists of an encoder, but instead of using a decoder afterwards, a SWAP test is used. 2 additional qubits are used on top of the 7 data qubits, an ancillary qubit and a reference qubit that is set to 0. The reference qubit is compared to the 7th trash qubit through a SWAP test. If the trash qubit differs from the reference qubit past a certain threshold, an anomaly is detected. Theoretically this is a less computationally expensive autoencoder than the quantum autoencoder. No decoder is required, and Instead of measuring all 7 qubits, we can simply measure the ancilla qubit.

Ansatz
PauliTwoDesign circuit:
The circuit consists of alternating layers of single-qubit Pauli rotations (X, Y, Z) and entanglement layers using controlled-Z (CZ) gates. The rotations are chosen randomly, and the entanglement pattern is fixed [2].



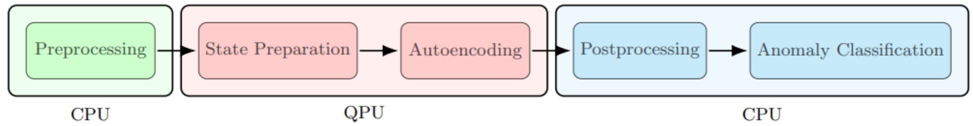
RealAmplitudes circuits with four entanglements (full, linear, circular, sca):
The circuit consists of alternating layers of Y-rotations and CNOT entanglement gates. We used predefined entanglement patterns named as full, linear, circular, sca [3].



Methodology

The method by which we implemented these autoencoders comprises of 5 steps:

1. Preprocessing
2. State Preparation
3. Autoencoding
4. Post Processing
5. Classification



Preprocessing

This step is used to process the data in a dataset. The datasets provided to us have continuous time series data which is hard to feed to the model. We converted this data into windows of a fixed size 2^K, where K denotes the number of qubits we are using (7 in our case). We move this window ahead one step at a time and this converts our continuous time series data into multiple windows.

Note that we do not normalize the data here as it will be done in the next step.

State Preparation

To convert each window to a quantum state, we use Amplitude Encoding
We chose Amplitude Encoding because it requires only log(N) qubits for N data points.
The process is deterministic and does not involve trainable parameters.

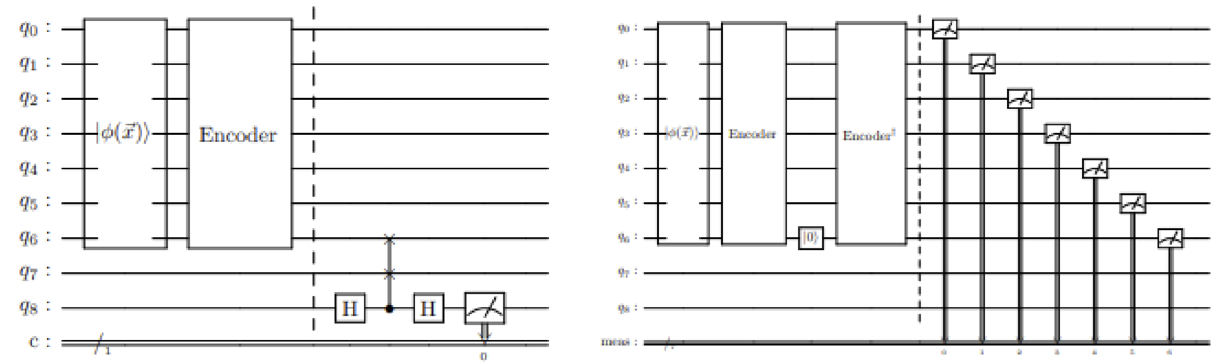
In this step, we convert each window to a corresponding quantum state. We use amplitude encoding here because it is very efficient in terms of size. It only requires log(N) qubits to represent N datapoints. Since the size of our window is 2^K, we can successfully convert each window to a quantum state with K qubits.

The formula used to convert a window $[w_{i0}, w_{i1}, \ldots, w_{i2^k-1}]$ to a quantum state using amplitude encoding is :

$$|\varphi_i\rangle = \frac{1}{\langle \vec{w}_i | \vec{w}_i \rangle} \vec{w}_i$$

Note that the steps mentioned till now are entirely deterministic and do not envolve any trainable parameters.

Autoencoding



(a) Trainable quantum autoencoder architecture (b) Trained quantum autoencoder architecture

After the state preparation, we can use either the trained autoencoder or trainable autoencoder to process the data. The encoder used in both implementations maps from the 2^7 Hilbert-Space to 2^6, essentially condensing information into qubits 0 to 5, while setting qubit 6 as close to the 0 state as possible [4].

The further qubit 6 is from the 0 state, the more likely an anomaly is present. This fact will be used for the post processing in the trainable autoencoder, while the trained autoencoder essentially “discards” qubit 6 by hard-setting it to the 0 state. This is because we are only interested in the reconstruction error for the MSE-based implementation.

Post Processing

MSE: The mean squared error is calculated based on the input state $|\varphi_i\rangle$ and output state $|\varphi'_i\rangle$, using the formula $\epsilon_i = (1/2K) \langle(\varphi_i - \varphi'_i)|(\varphi_i - \varphi'_i)\rangle$ [4].

SWAP Test: For the SWAP test, we generate the vector $\vec{\epsilon} = [\epsilon_i \ldots \epsilon_i$ of the same length as the number of time windows, and ϵ_i is the SWAP test measurement for each time window [4]. In simple terms, we simply compile all of the SWAP test measurements instead of doing MSE calculations.

After the error data is compiled, we used moving average is used to eliminate potential outliers in the data, where we slide a fixed window size on the processed data ϵ.

Classification

Finally, we use the post processed data to identify anomalies, where if the error is above a predefined threshold, it is classified as an anomaly.

Software Results

We trained the QAE with dataset 028, 054, 099, 118, 138, and 176 with the five ansatz and MSE based and Swap-Test based detection. The dataset indicates information in their fine name as “(Dataset number)_(name)_(from 1 to this number is training data)_(begin anomaly)_(end anomaly).txt”. For example, “028_UCR_Anomaly_DISTORTEDInternalBleeding17_1600_3198_3309.txt” means that its dataset number is 028, training from 1 to 1600, anomaly starts from 3198 to 3309.

We found that MSE based auto encoding worked better than SWAP based, while the paper argues that they found that SWAP based auto encoding was superior to MSE based.

Pictures of the results can be found in the **Appendix.

Ease-of-reproducibility and Reproducibility Issues

Ease of Reproduction

Due to the quality and thoroughness of the methodology steps, replicating the software described in the paper was simple.

Accessible and well-documented dataset
The authors used the publicly available and widely recognized UCR Time Series Anomaly Archive dataset. This archive includes datasets from various domains—such as medicine, sports, and robotics. Each dataset is well-documented, containing information about training and anomaly ranges.

Clear methodological explanation
The paper clearly explains the 5 methodology steps, ranging from preprocessing and state preparation to autoencoding and postprocessing. Each step is supported by explanatory figures and thorough descriptions, making it easy to follow. In particular, the decision to standardize window sizes (e.g., 128) and the rationale behind amplitude encoding are justified properly.

Detailed ansatz descriptions
The paper does a good job of explaining each ansatz it uses. It includes the type of entanglement, how many parameters each version has, and what each design is trying to achieve. This made it much easier for us to rebuild the circuits ourselves. Since they used standard Qiskit ansätze like RealAmplitudes and PauliTwoDesign, we were able to use the same ones directly without having to recreate them from scratch.

Limitations of reproduction

The following are some challenges that we faced while reproducing the paper:

Computational Limitation
The paper works with 5 types of ansatz, and each ansatz can have 9 different numbers of parameters. Running each configuration multiple times on 6 datasets gives us over 600 hours of training time on a normal laptop. Due to this limitation, we have not replicated all the results given in the paper, but rather we have chosen a few configurations to work with(these can be found in the results section).

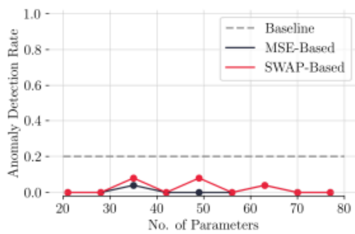
Lack of detail in classification
While the paper mentions that a threshold is used for anomaly detection, it does not explain how this threshold should be calculated. Using a different threshold than the one used in the paper may affect the consistency of the results.

Two separate classification methods
The paper uses two distinct approaches for anomaly classification: MSE and Swap-Test measurements. This makes reproduction a bit more work, since both methods need to be implemented and tested in the software.

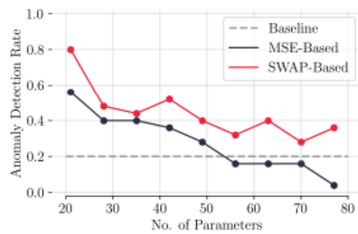
Validity of the Paper

no.																			
28	PauliTwoDesign	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RealAmplitude - Circular	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RealAmplitude - Full	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RealAmplitude - Linear	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RealAmplitude - SCA	0.00	0.00	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.00	**0.40**	0.00	**0.40**	0.00	0.20	0.00	0.00
	Classical Autoencoder					0.20									0.20				
54	PauliTwoDesign	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.80**	**0.80**	**0.80**	0.20	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	RealAmplitude - Circular	0.20	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	0.20	0.00	0.20	0.00	0.00	0.00	0.00
	RealAmplitude - Full	0.00	0.00	0.20	0.00	0.20	0.00	0.00	0.00	0.00	**1.00**	0.00	0.20	**0.40**	**0.40**	0.20	**0.60**	**0.40**	**0.60**
	RealAmplitude - Linear	**0.80**	**0.80**	**0.60**	**0.60**	0.20	0.00	0.00	0.00	0.00	**1.00**	**0.80**	**0.80**	**0.60**	0.20	**0.40**	**0.40**	0.00	0.20
	RealAmplitude - SCA	**0.80**	0.20	0.00	0.20	0.00	0.00	0.00	0.00	0.00	**0.80**	**0.60**	0.00	**0.60**	0.20	0.00	0.00	0.00	0.00
	Classical Autoencoder					0.20									0.20				
99	PauliTwoDesign	0.00	0.00	0.00	0.00	0.00	**0.40**	0.00	0.00	0.00	0.00	0.00	0.20	0.00	0.00	**0.40**	0.00	0.00	0.00
	RealAmplitude - Circular	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.20	**0.40**
	RealAmplitude - Full	0.00	0.00	0.00	0.20	0.20	0.00	0.20	0.00	**0.40**	0.20	0.00	0.00	0.20	0.20	0.00	0.00	0.20	0.20
	RealAmplitude - Linear	0.00	0.20	0.00	0.00	0.20	0.20	0.20	0.00	0.20	0.00	0.00	0.00	0.00	0.20	0.20	**0.40**	0.20	0.20
	RealAmplitude - SCA	0.00	0.00	0.00	0.00	0.20	0.00	0.00	**0.40**	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20
	Classical Autoencoder					0.20									0.20				
118	PauliTwoDesign	0.00	0.00	0.00	0.00	0.00	0.20	0.00	**0.20**	0.00	**0.40**	**0.40**	**0.40**	**0.60**	**0.60**	**0.60**	0.20	**1.00**	**0.60**
	RealAmplitude - Circular	0.20	**0.40**	0.00	0.20	**0.40**	0.00	0.20	0.00	0.20	**0.80**	**0.80**	**0.80**	**0.60**	**1.00**	**1.00**	**0.80**	**0.60**	**1.00**
	RealAmplitude - Full	0.00	**0.80**	**0.60**	0.20	**0.40**	**0.40**	0.20	**0.40**	0.20	**0.80**	**1.00**	**1.00**	**0.80**	**0.40**	**0.80**	**0.60**	**0.80**	**0.80**
	RealAmplitude - Linear	0.00	0.00	0.00	0.20	0.40	0.20	**0.80**	0.20	0.20	0.20	**0.60**	0.20	**0.60**	**0.80**	**0.60**	**0.80**	**0.80**	**0.80**
	RealAmplitude - SCA	0.00	0.20	**0.60**	0.20	0.00	0.20	0.20	0.00	0.00	**0.60**	**0.60**	**0.60**	**0.80**	0.00	**0.80**	**0.80**	**0.60**	**1.00**
	Classical Autoencoder					0.20									0.20				
138	PauliTwoDesign	0.00	0.00	0.00	**0.40**	**0.40**	0.20	**0.40**	0.00	0.00	0.20	0.00	0.00	**0.40**	**0.80**	0.20	0.00	0.20	0.00
	RealAmplitude - Circular	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	**0.40**	0.20	0.00	0.00	0.20
	RealAmplitude - Full	0.00	0.00	0.00	0.20	0.20	0.00	0.00	0.00	0.20	0.20	0.00	0.00	**0.40**	0.20	0.00	0.20	0.00	0.20
	RealAmplitude - Linear	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.20	0.20	0.00	0.00	0.00	**0.40**	0.20	0.20	0.00
	RealAmplitude - SCA	0.00	0.00	**0.60**	0.00	0.20	0.20	**0.40**	0.00	0.00	**0.40**	0.00	**0.60**	0.20	**0.60**	**0.40**	**0.60**	0.20	**0.40**

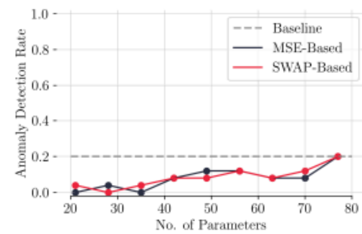
- The paper constantly asserts that the quantum autoencoder triumphs over the classical one. However, if we take a look at multiple datasets, we will come to a different conclusion. While the authors come to the conclusion from Table 2 that quantum consistently outperforms the classical baseline of 0.2, we see an issue. Upon closer examination, several datasets perform consistently at 0, if the paper wants to cherry pick their result,s we do not see why they couldn’t cherry pick their datasets. Performance varied across datasets and ans¨atze, highlighting the importance of ansatz selection. Swap-Test-based anomaly detection appeared more resilient at higher loss levels, while MSE-based detection excelled at lower loss The “superior” performance highlighted in bold is often for a particular ansatz and a specific number of trainable parameters. Based on these results, we cannot come to the conclusion that quantum autoencoders outperform classical; rather, it seems that classical is better if not equal in the majority of cases. - This doesn’t say much either as the baseline is 0.2/1 a 20% success threshold.



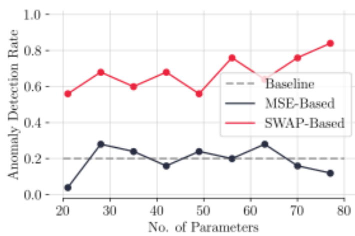
(a) Dataset no. 28



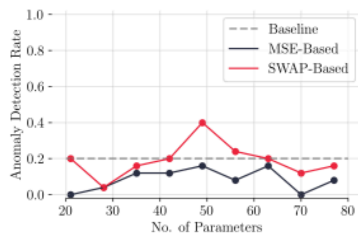
(b) Dataset no. 54



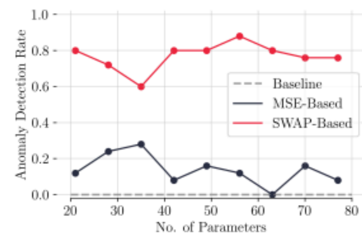
(c) Dataset no. 99



(d) Dataset no. 118



(e) Dataset no. 138



(f) Dataset no. 176

. "*The baseline corresponds to the conventional deep learning-based autoencoder setup as described in Section 4.2. Exhibiting subpar performance on a specific dataset compared to the baseline deep learning-based autoencoder does not imply that quantum autoencoders are intrinsically inferior for that dataset. Rather, it indicates that the average performance across all tested ansätze is lower. There may still exist a particular ansatz that surpasses the baseline, as demonstrated in Table 2.*"[5]

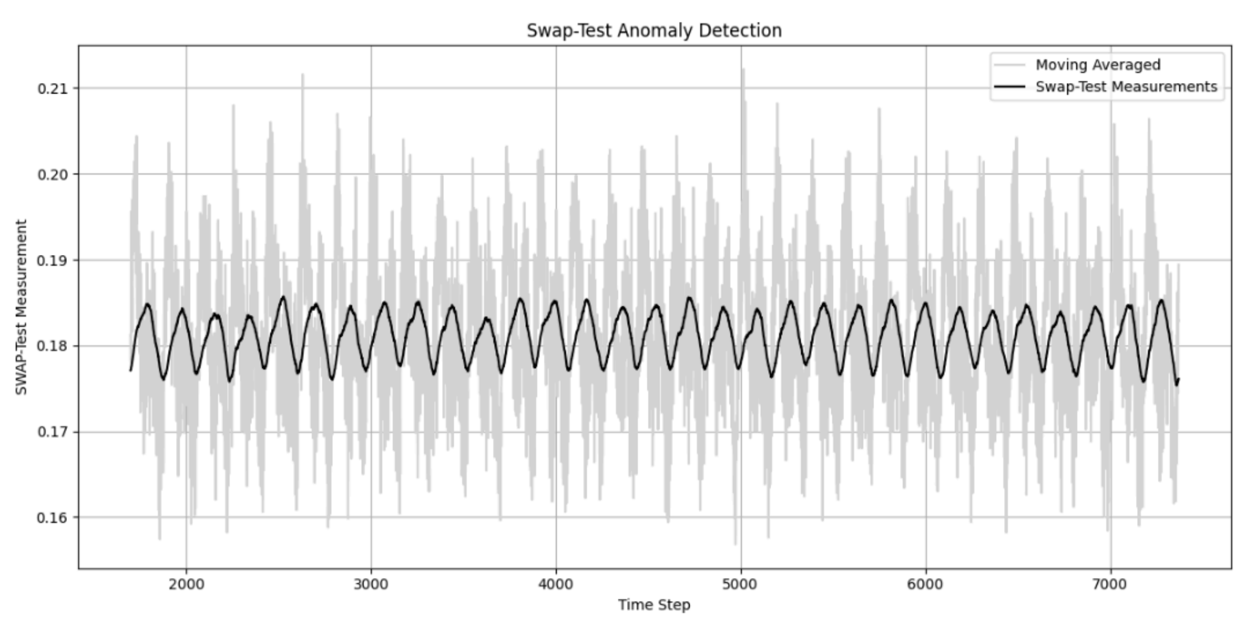
Just based on the surface, the detection method of the quantum method falls below the threshold multiple times. Besides that issue, the paper attempts to display the data as a conglomerate of all 5 different *ansätze* methods, providing a pretty big bias for quantum autoencoders. While this would be impressive if all methods were around the same success rate with the baseline threshold of 0.2 or even 0, it is not hard for one circuit of one set of parameters of MSE or SWAP-based methodology to lift the average above 0.2.

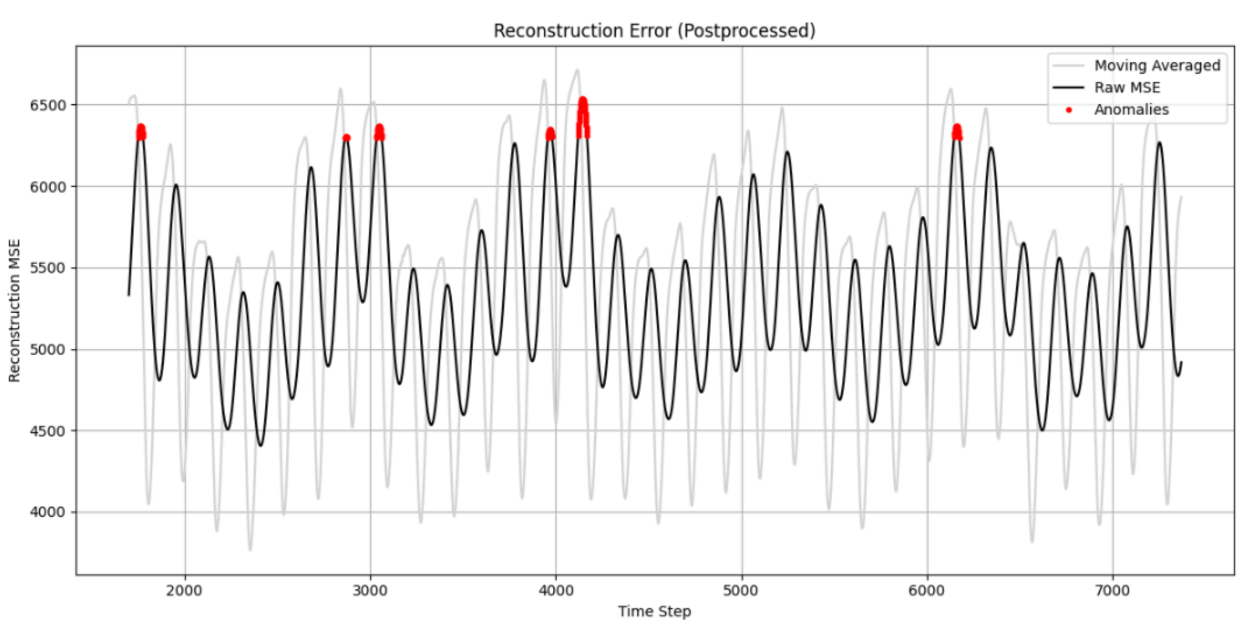
In conclusion, the paper's results of consistent outperformance are believed to have been influenced by the best-performing quantum configuration and biased display of data. Further, they seem to constantly downplay the shortcomings of the results. This can be seen by their attempts to average the quantum results on the datasets. This is clearly a rather incorrect blanket statement, as there is almost no consistent performance across one circuit and one set of parameters. Therefore, the reader should interpret the claim of consistent quantum superiority with caution and consider the context-dependent nature of the observed out-performance..

Appendix

Dataset 028: 028_UCR_Anomaly_DISTORTEDInternalBleeding17_1600_3198_3309.txt

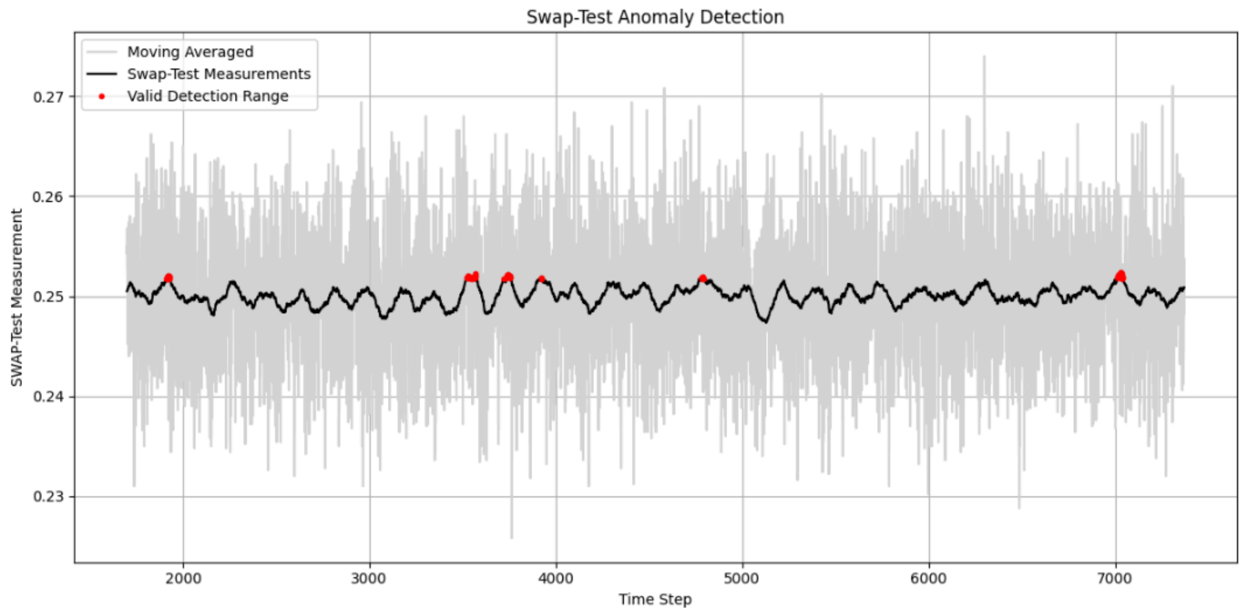
PauliTwoDesign:

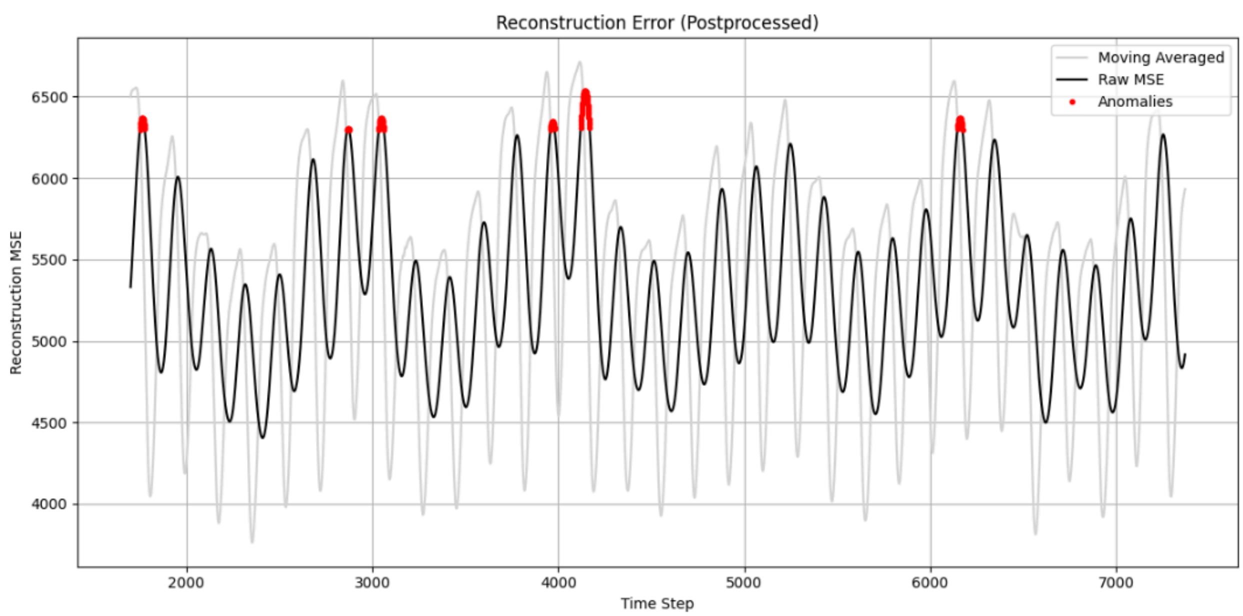




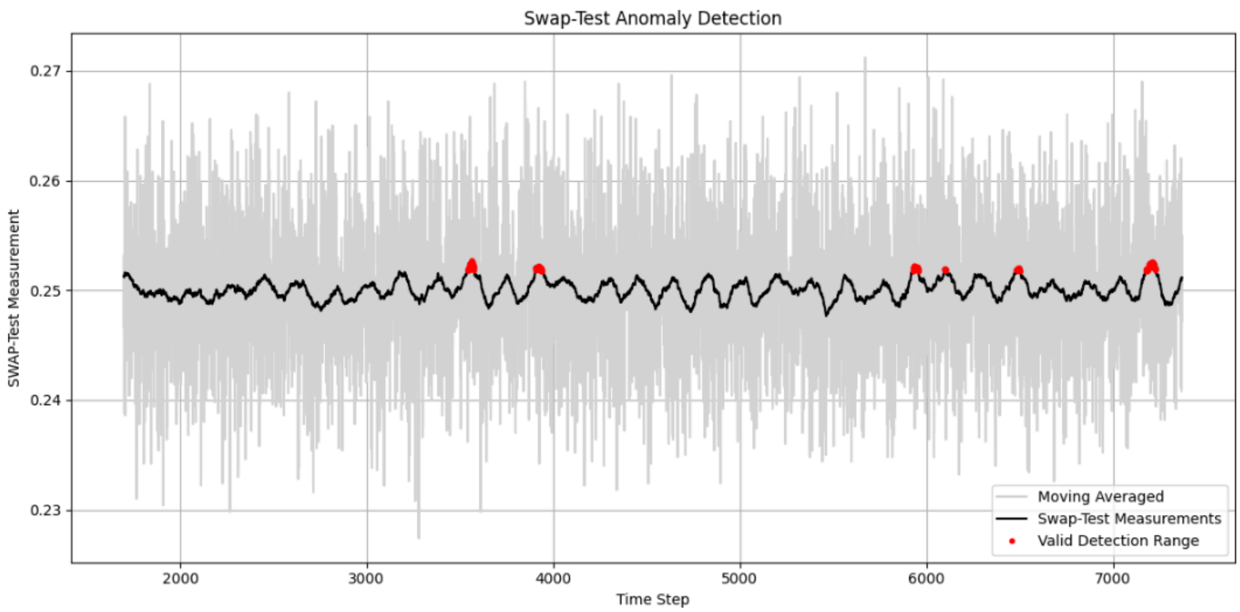
RealAmplitudes:

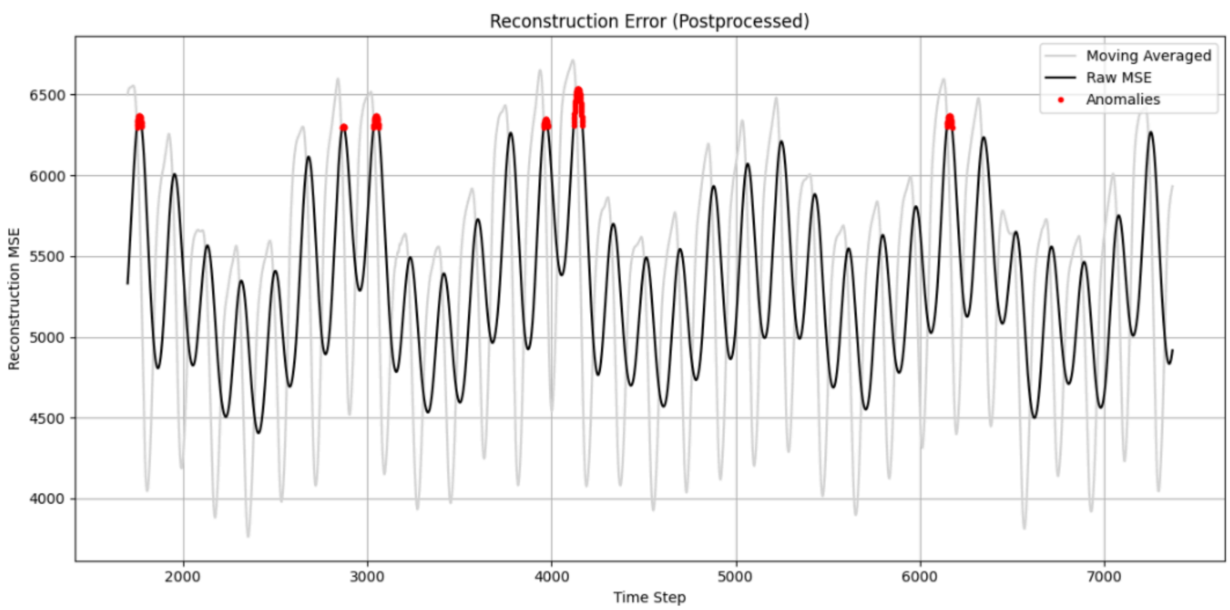
entanglement=”circular”



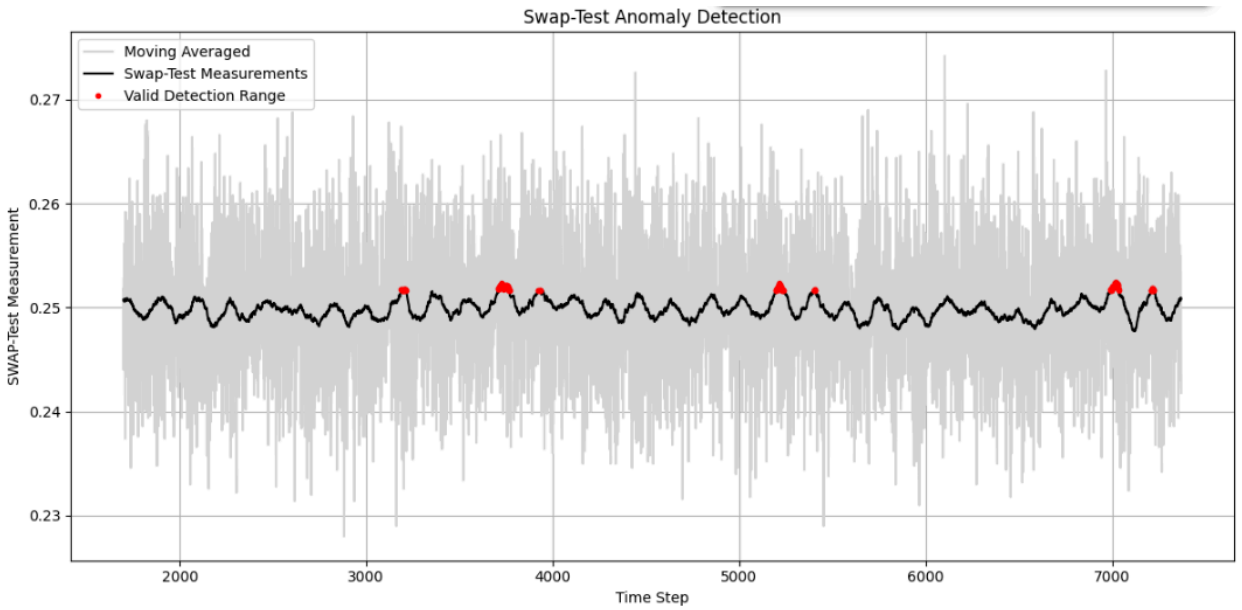


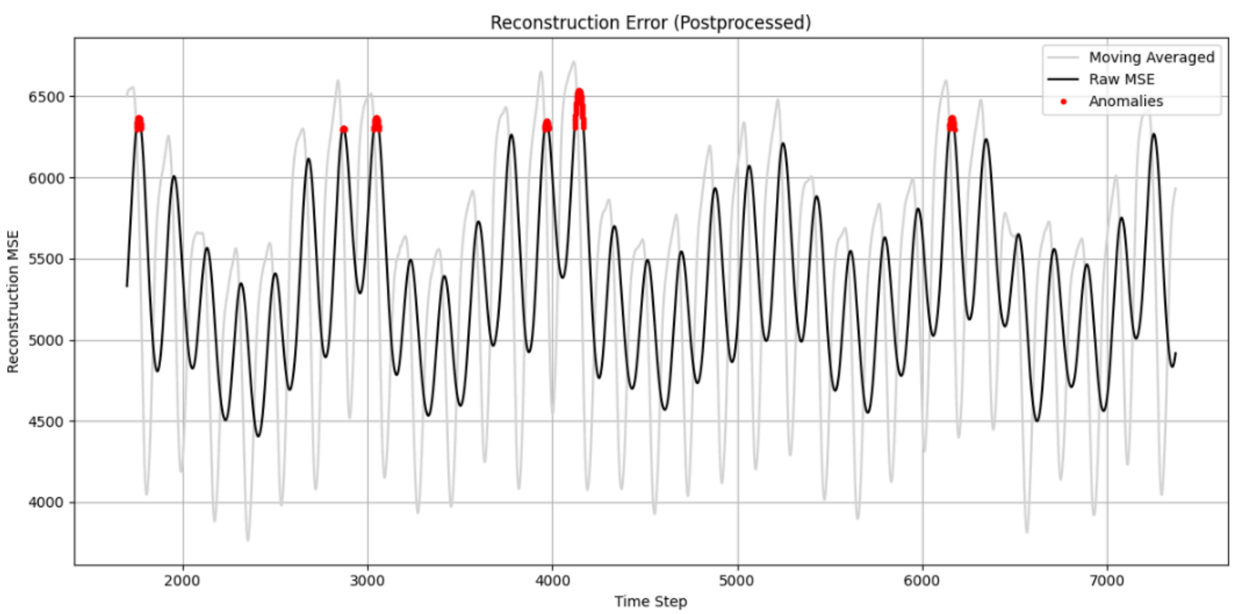
entanglement="full"



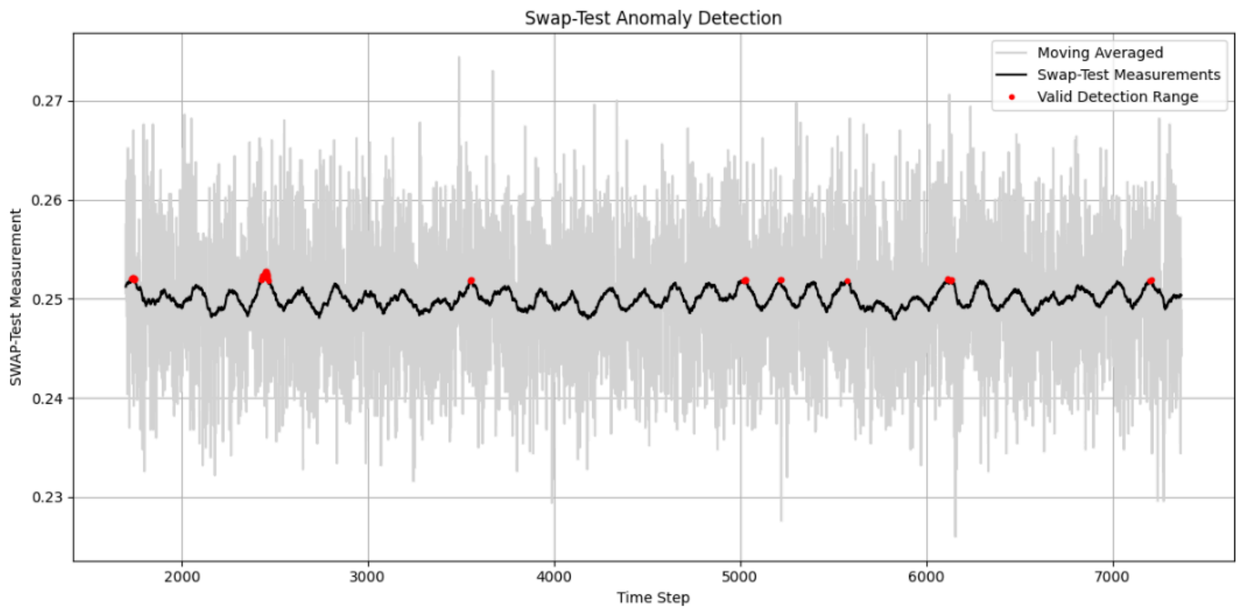


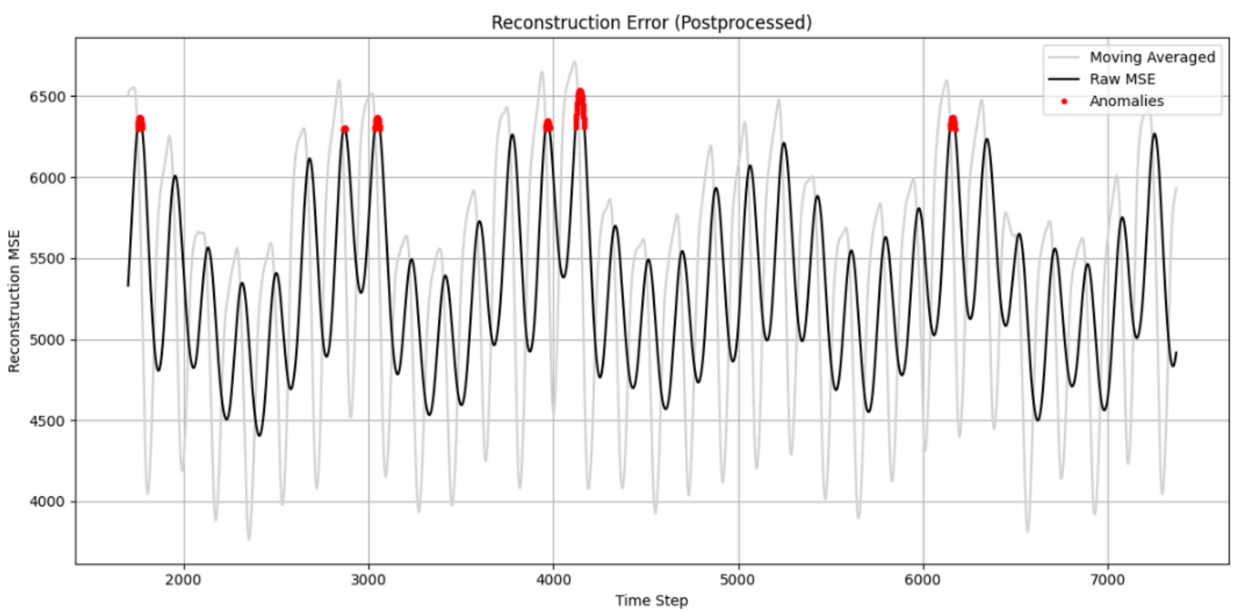
entanglement="linear"





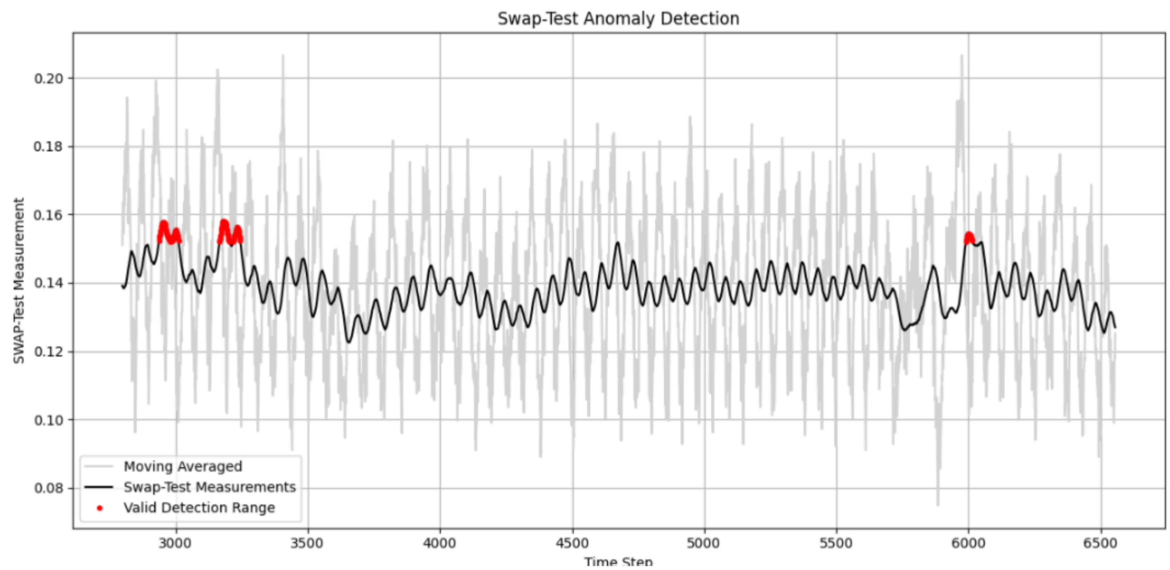
entanglement=”sca”

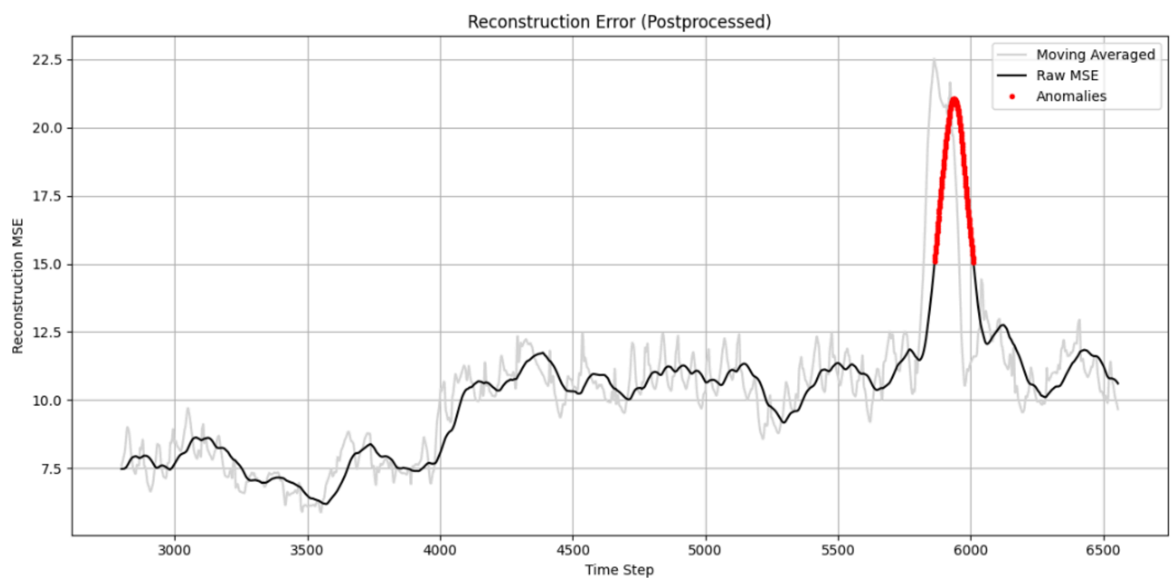




Dataset 054: 054_UCR_Anomaly_DISTORTEDWalkingAceleration5_2700_5920_5979.txt

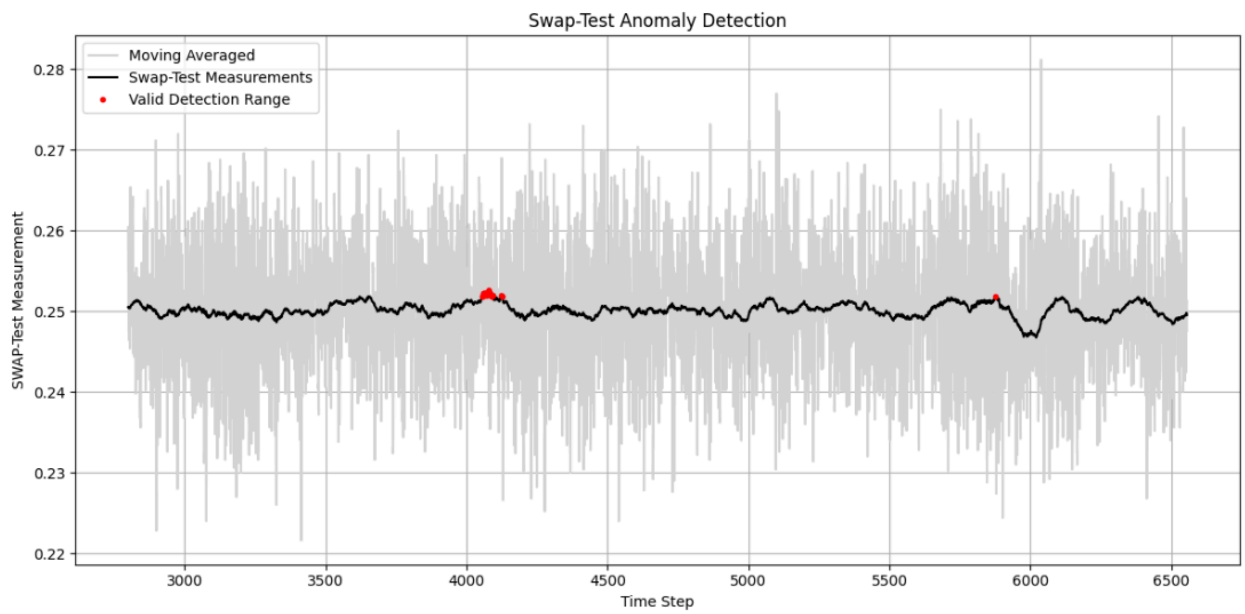
PauliTwoDesign:

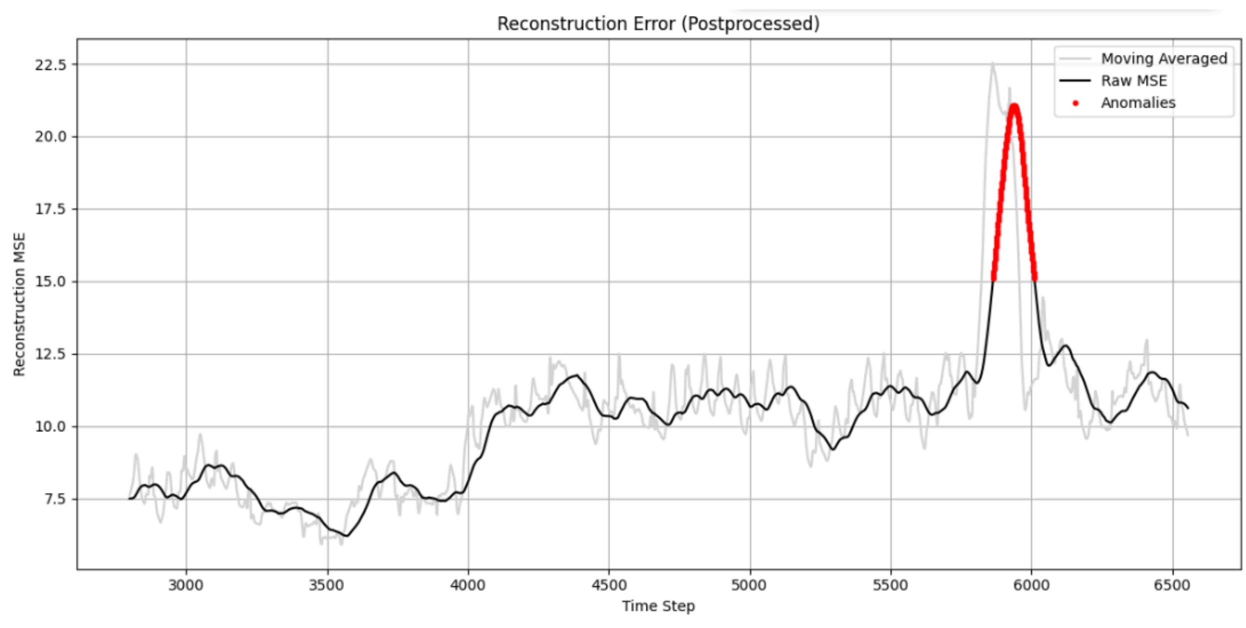




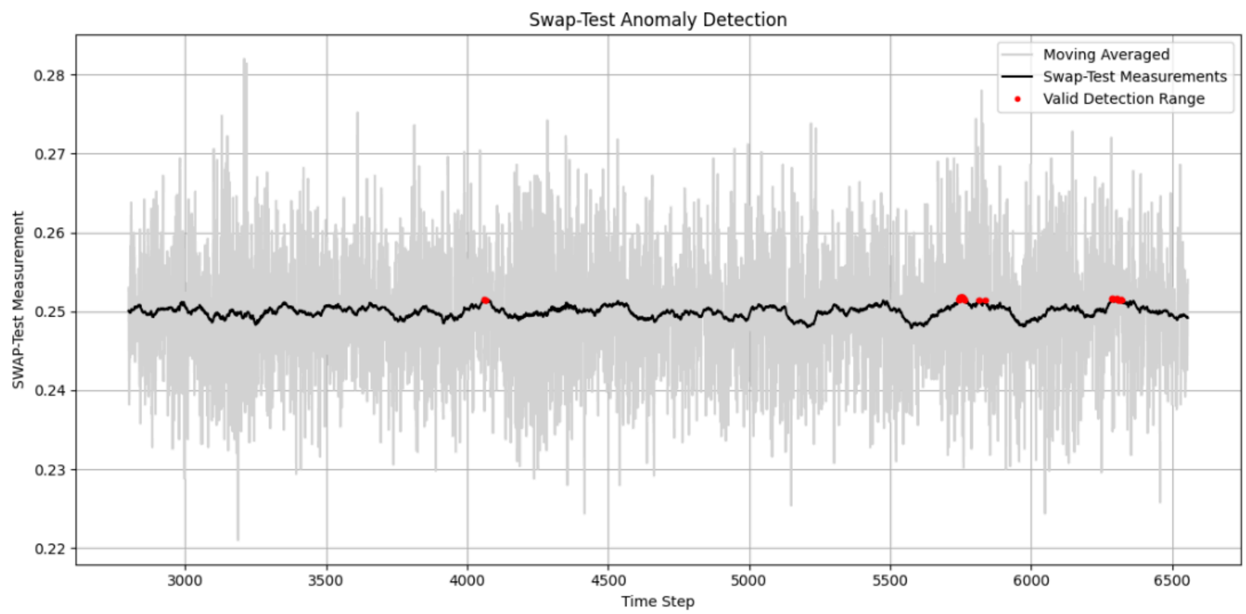
RealAmplitudes:

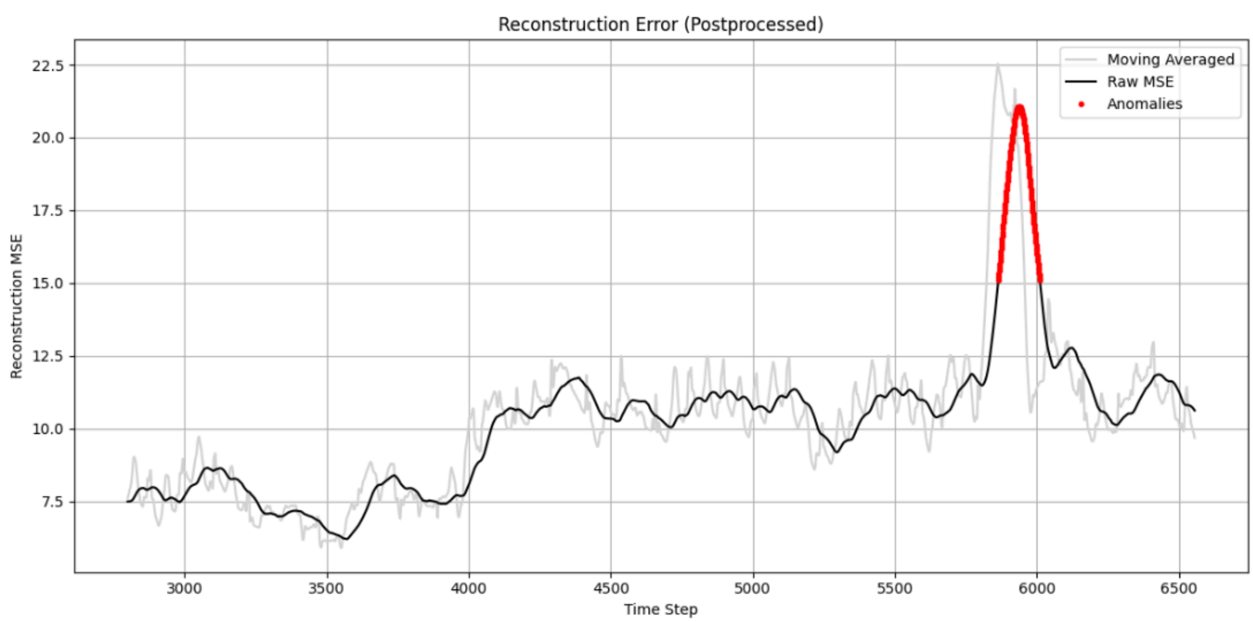
entanglement="circular"



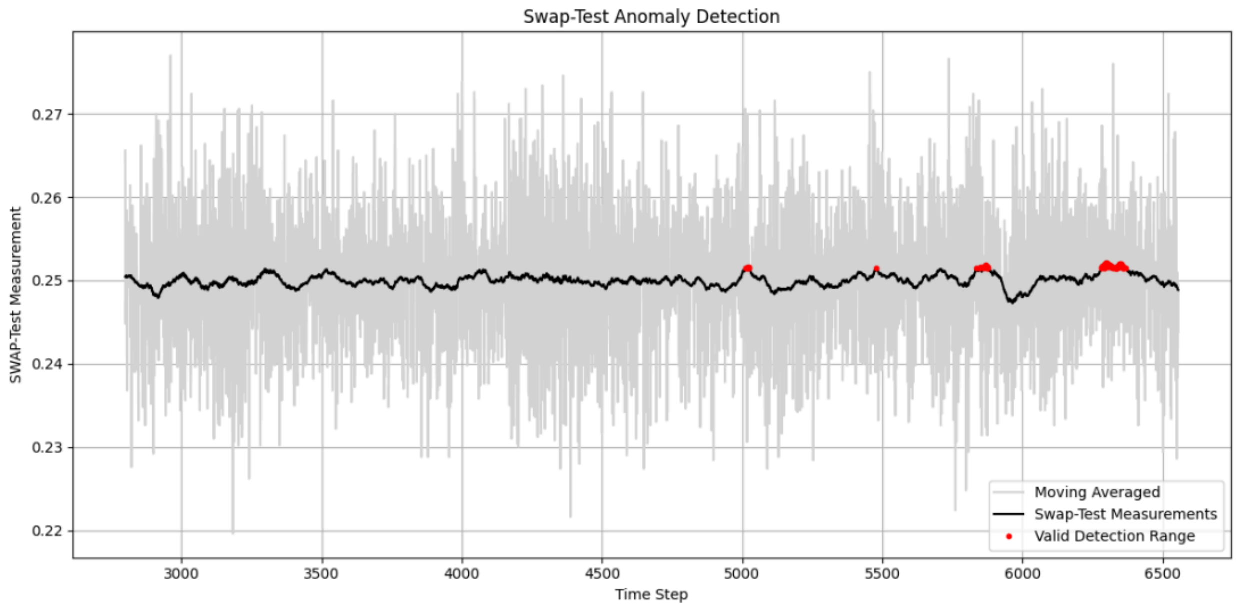


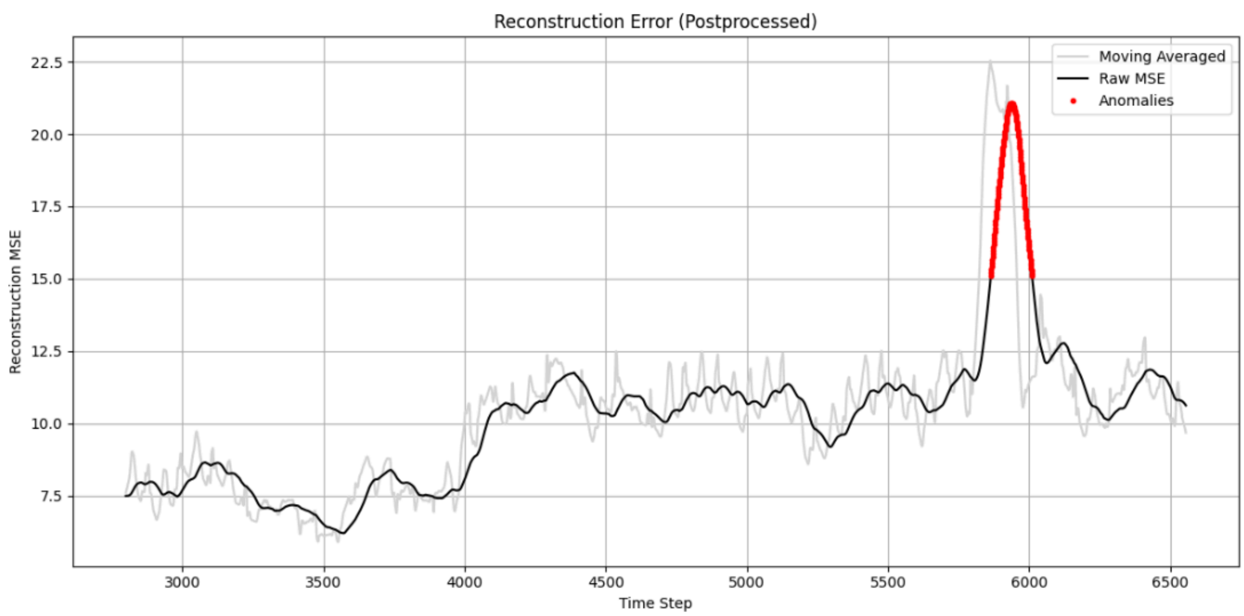
entanglement="full"



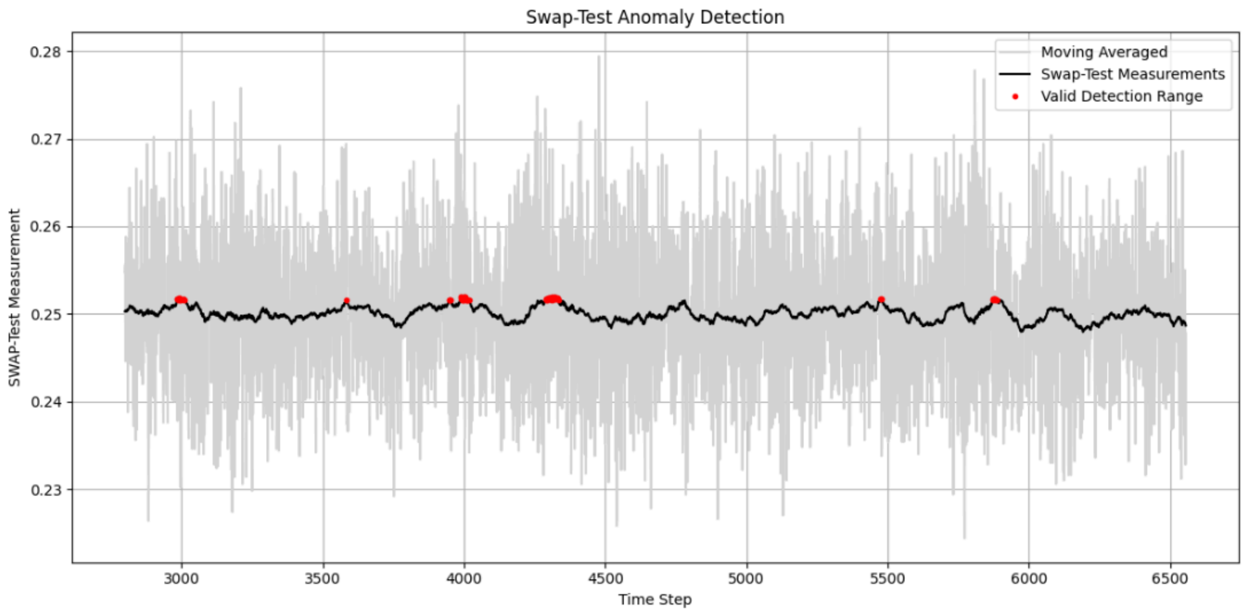


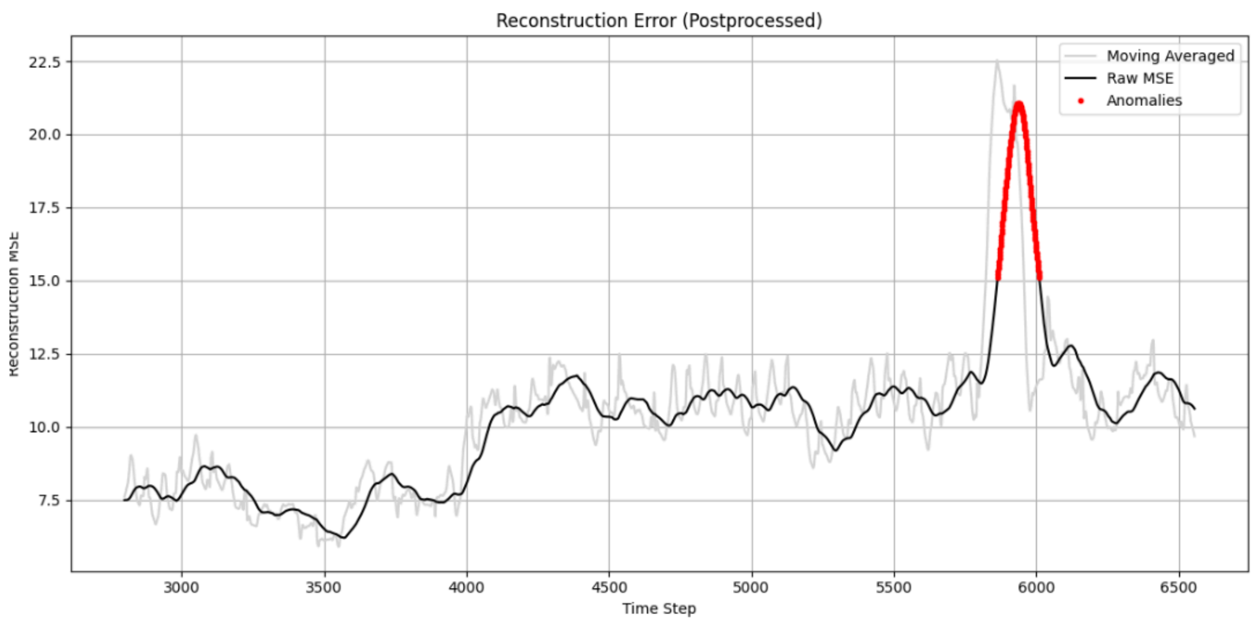
entanglement="linear"





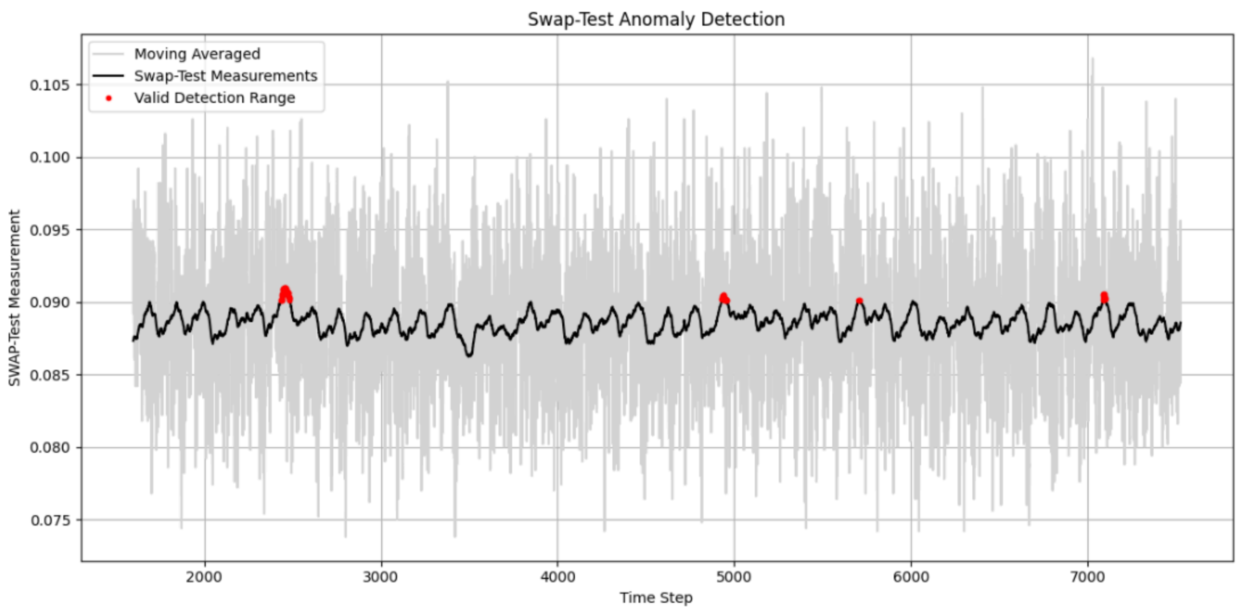
entanglement=”sca”

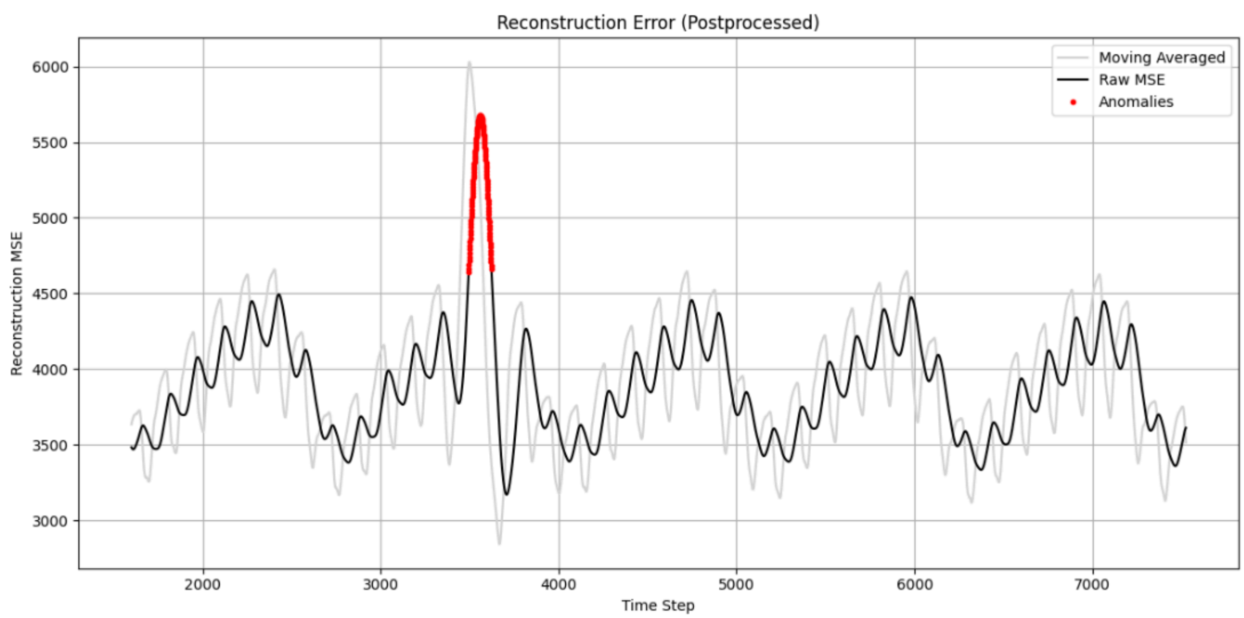




Dataset 099: 099_UCR_Anomaly_NOISEInternalBleeding6_1500_3474_3629.txt

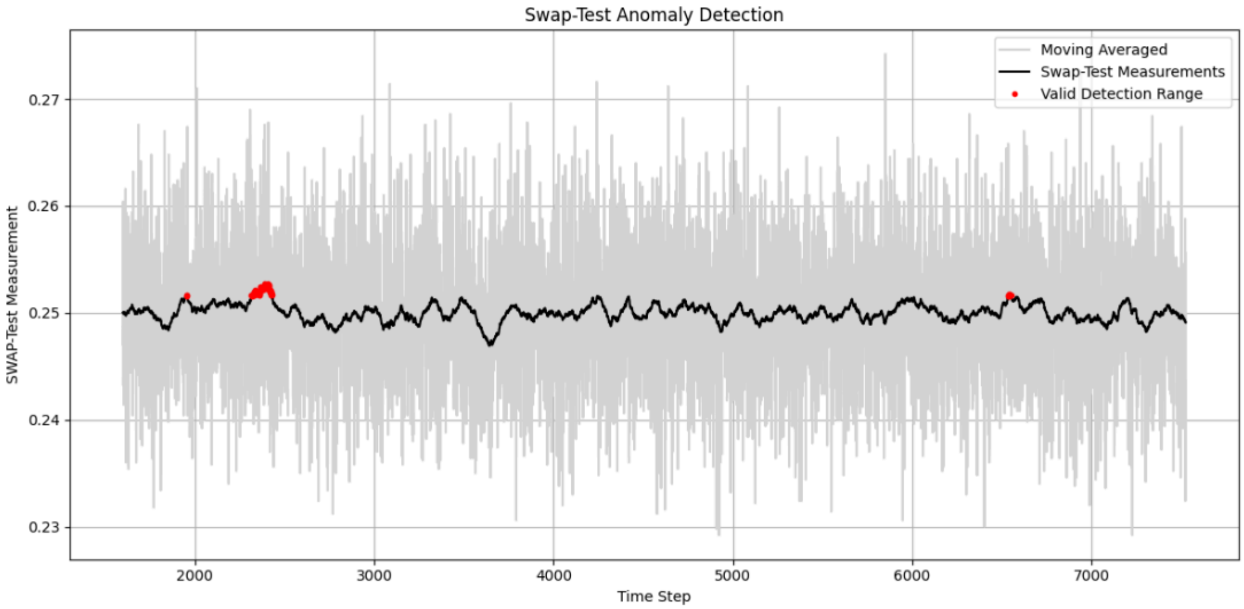
PauliTwoDesign:

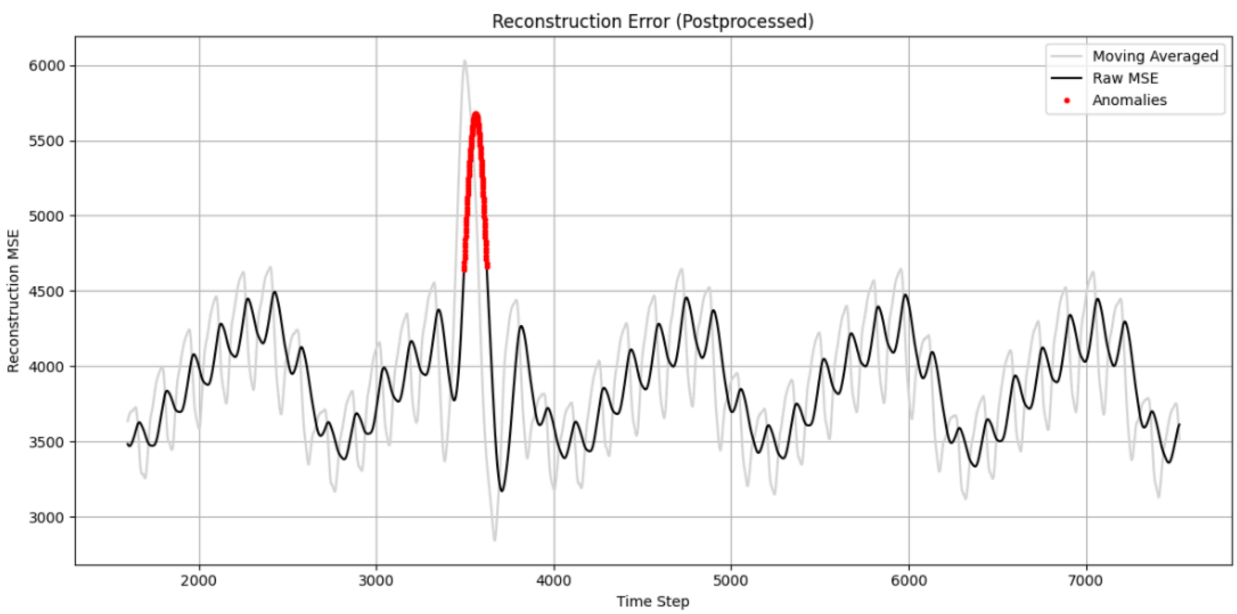




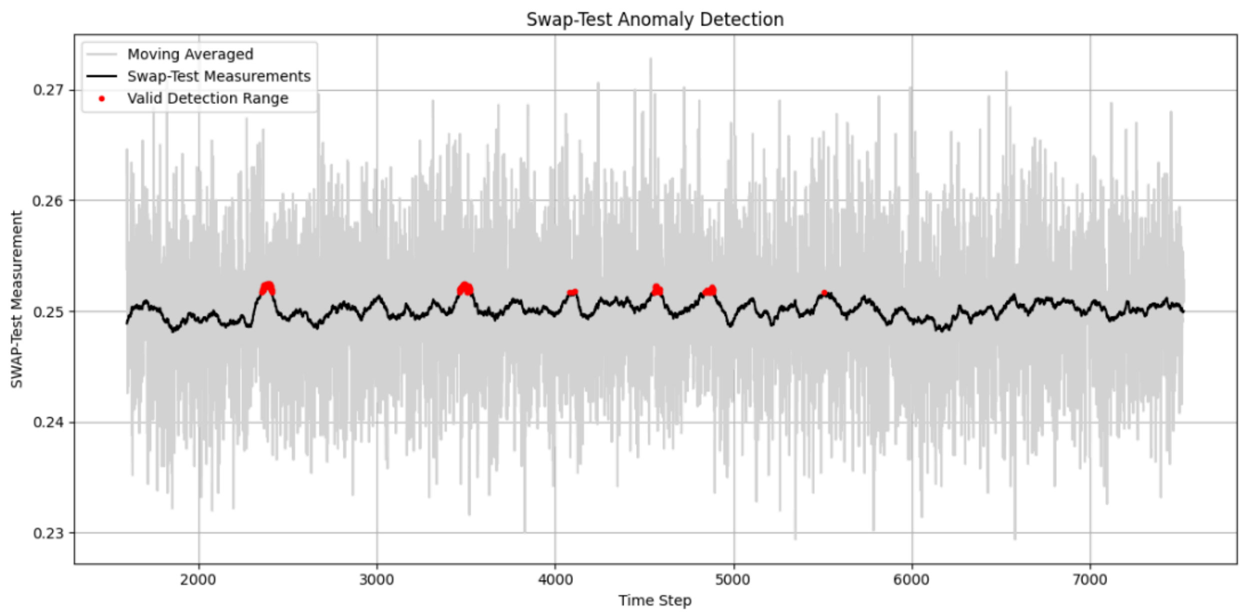
RealAmplitudes:

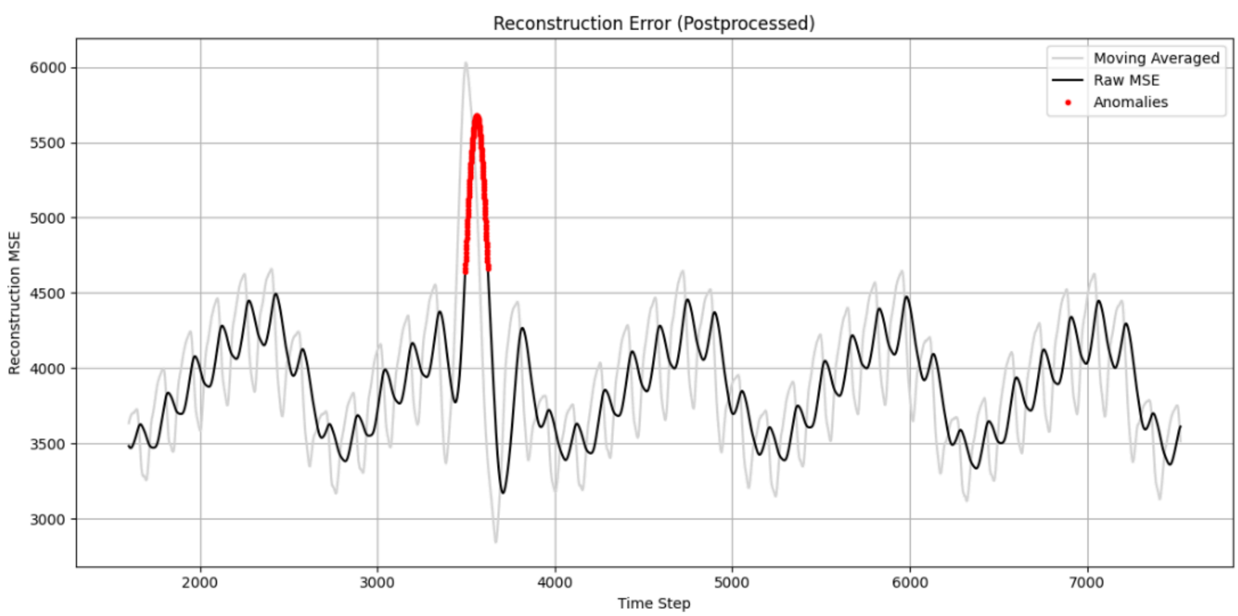
entanglement=”circular”



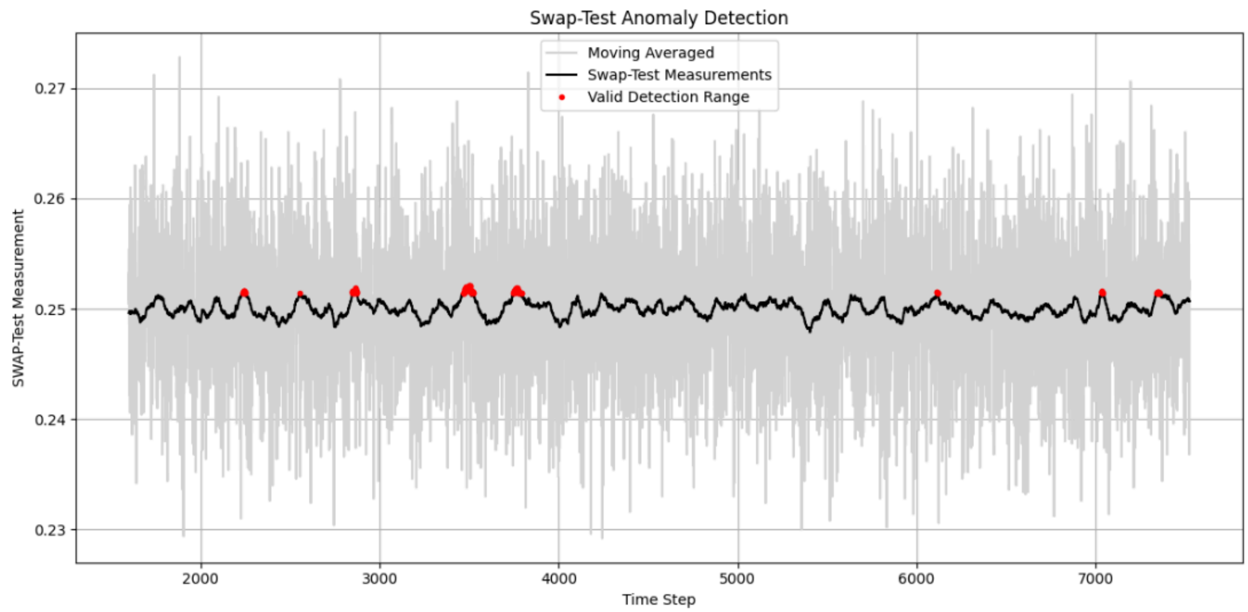


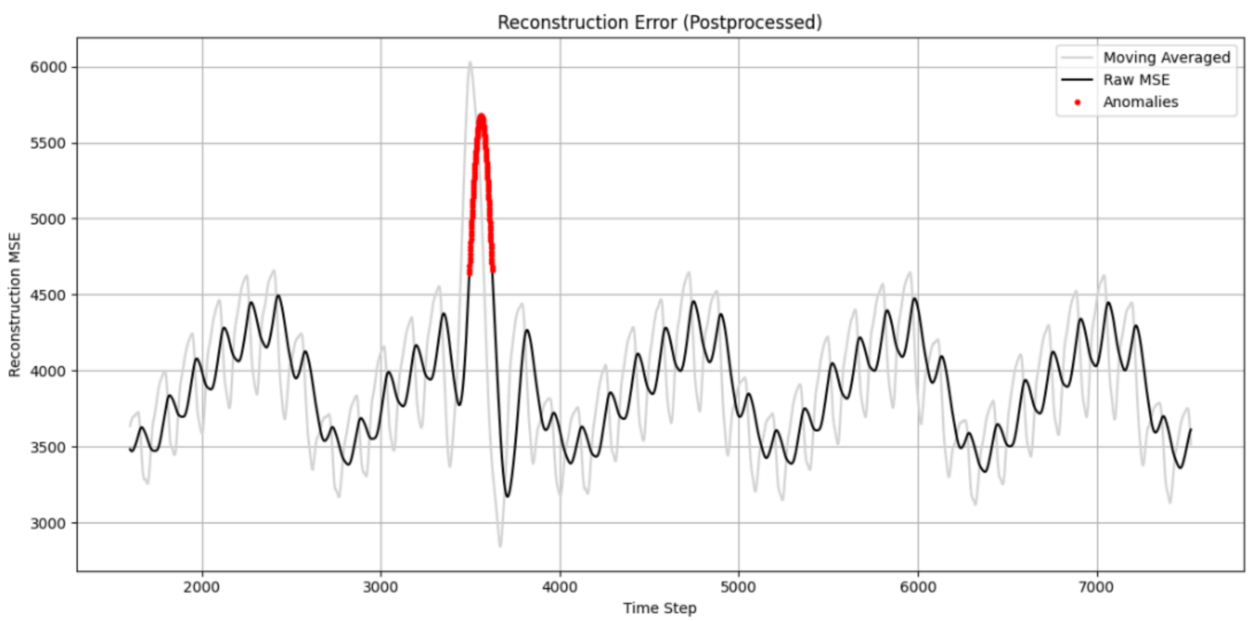
entanglement="full"



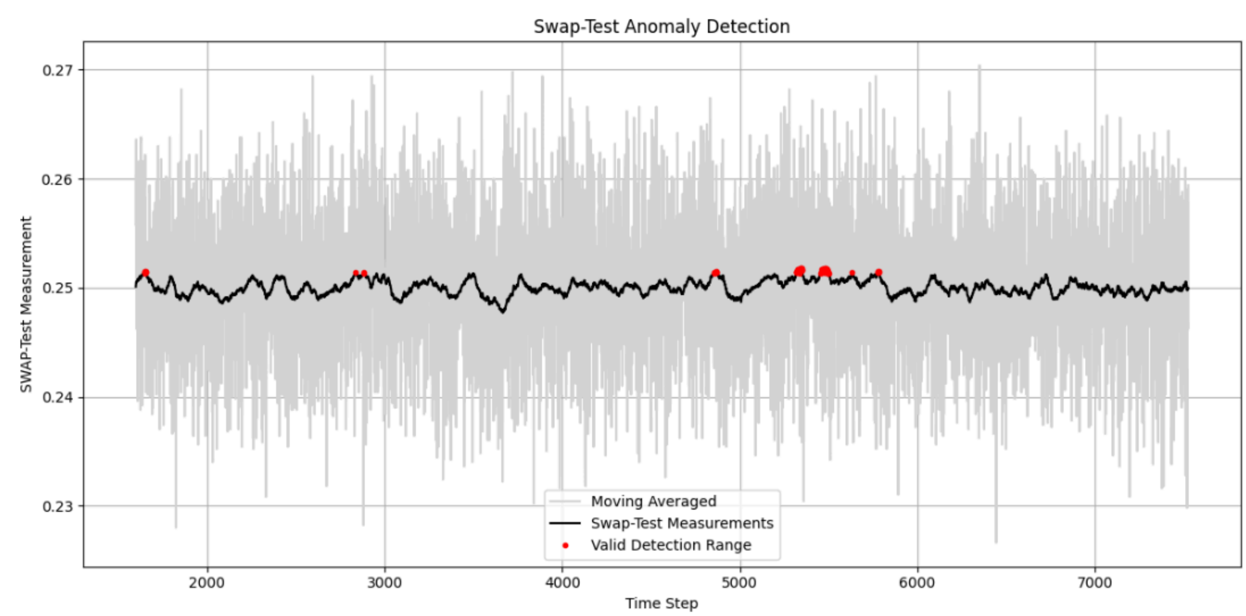


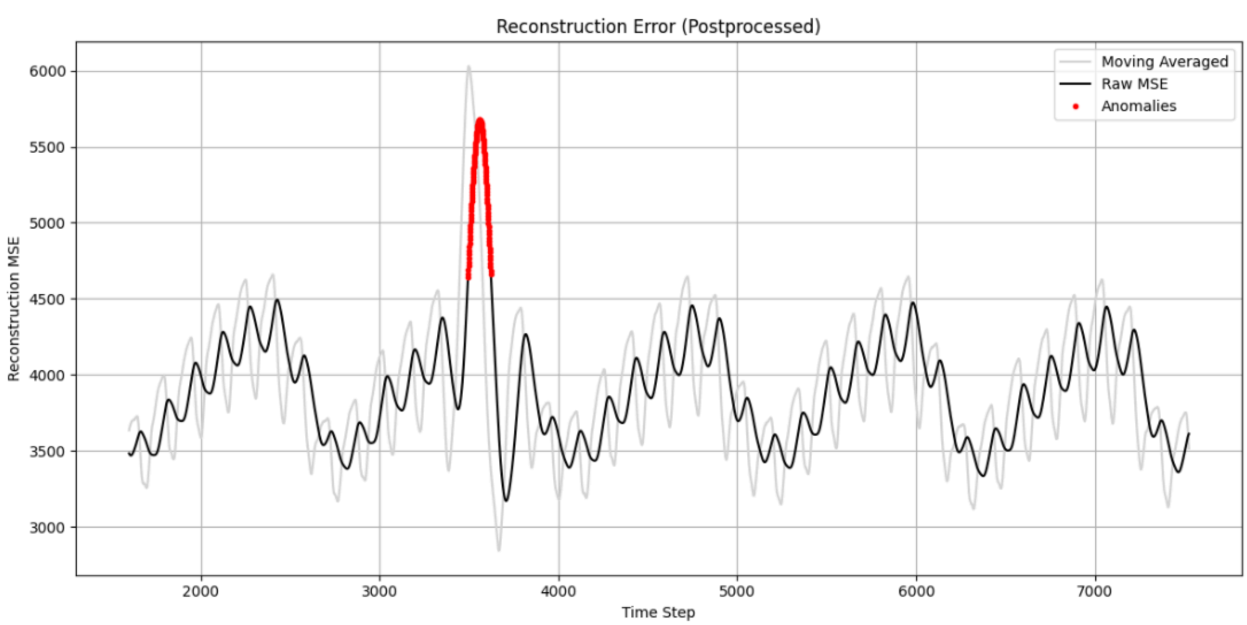
entanglement="linear"





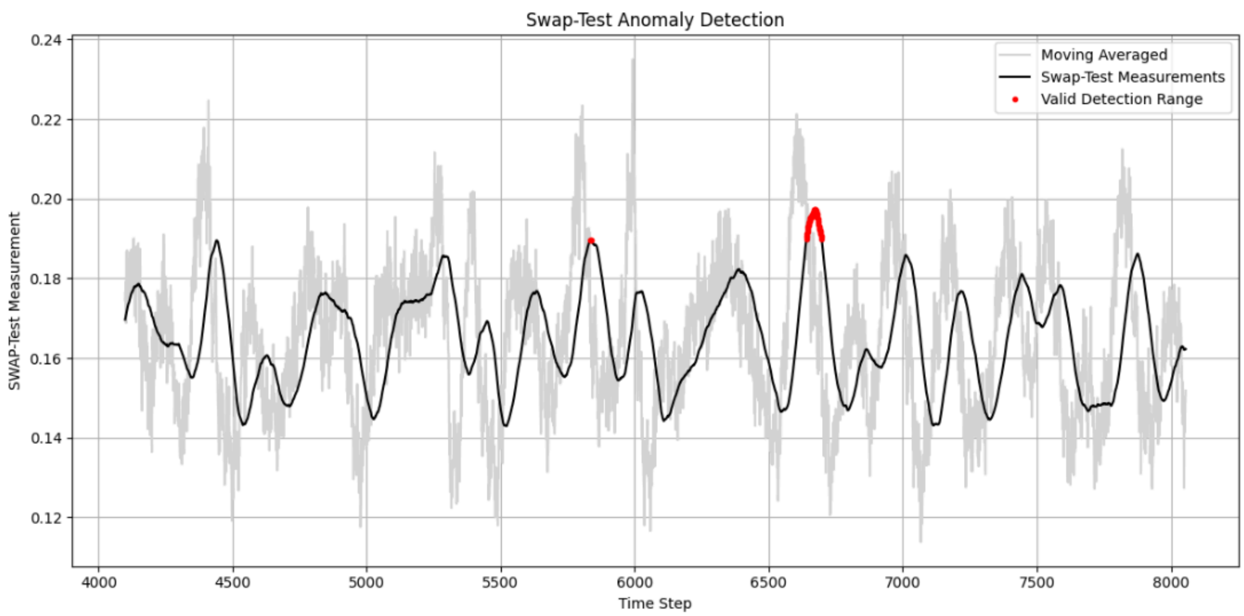
entanglement="sca"

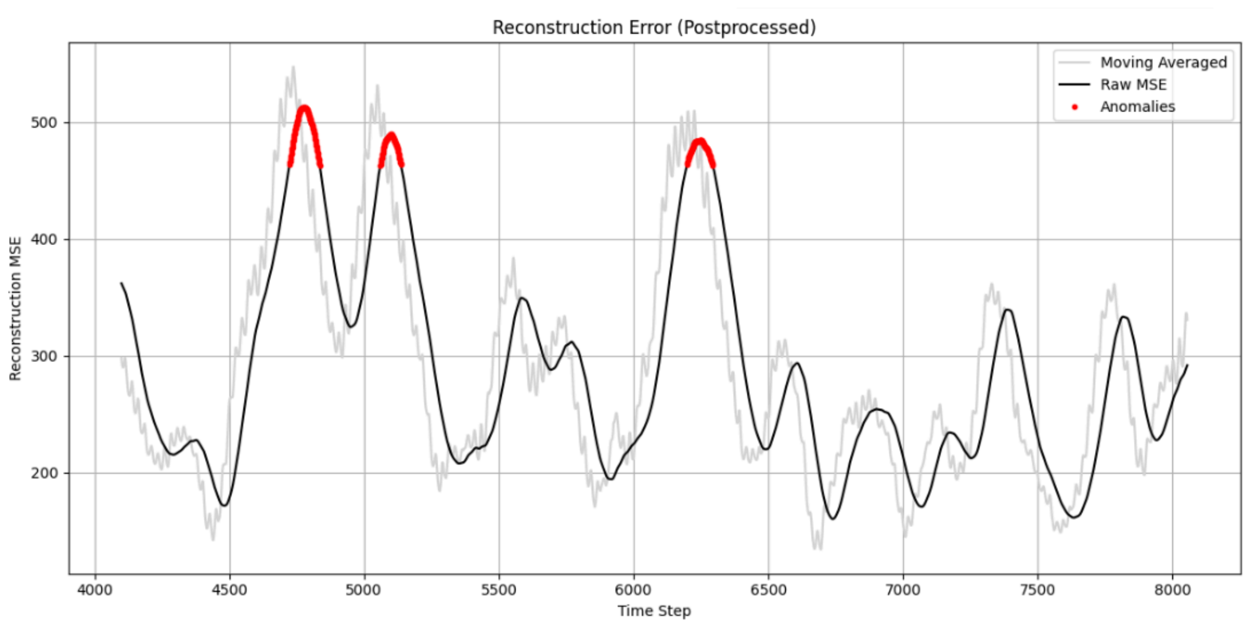




Dataset 118: 118_UCR_Anomaly_CIMIS44AirTemperature6_4000_6006_6054.txt

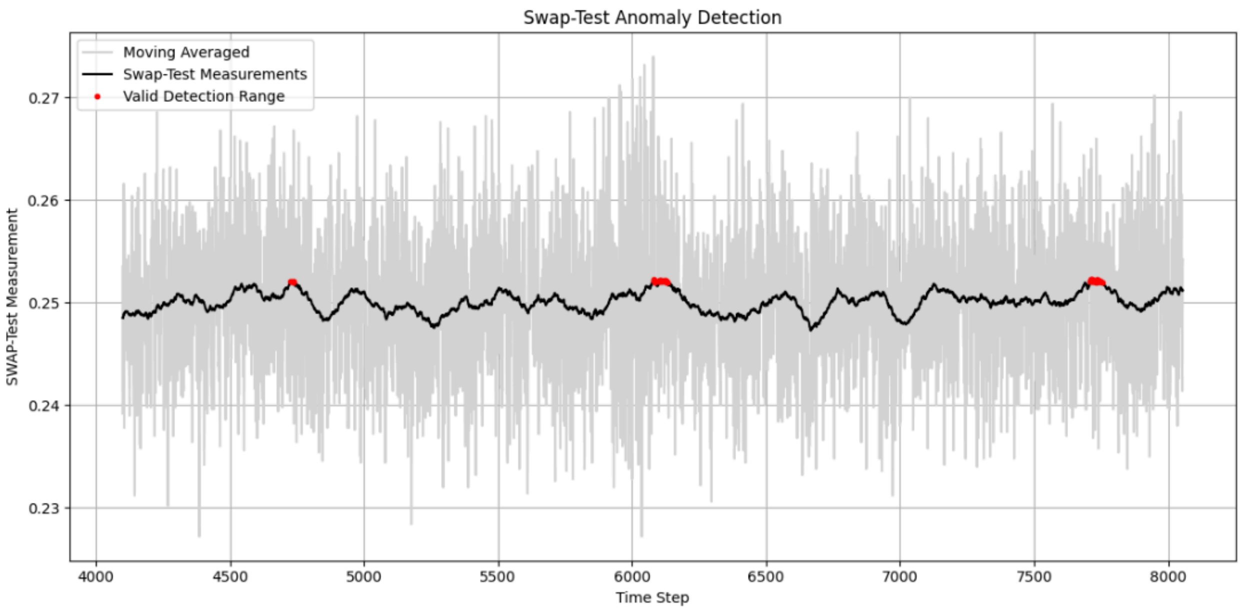
PauliTwoDesign:

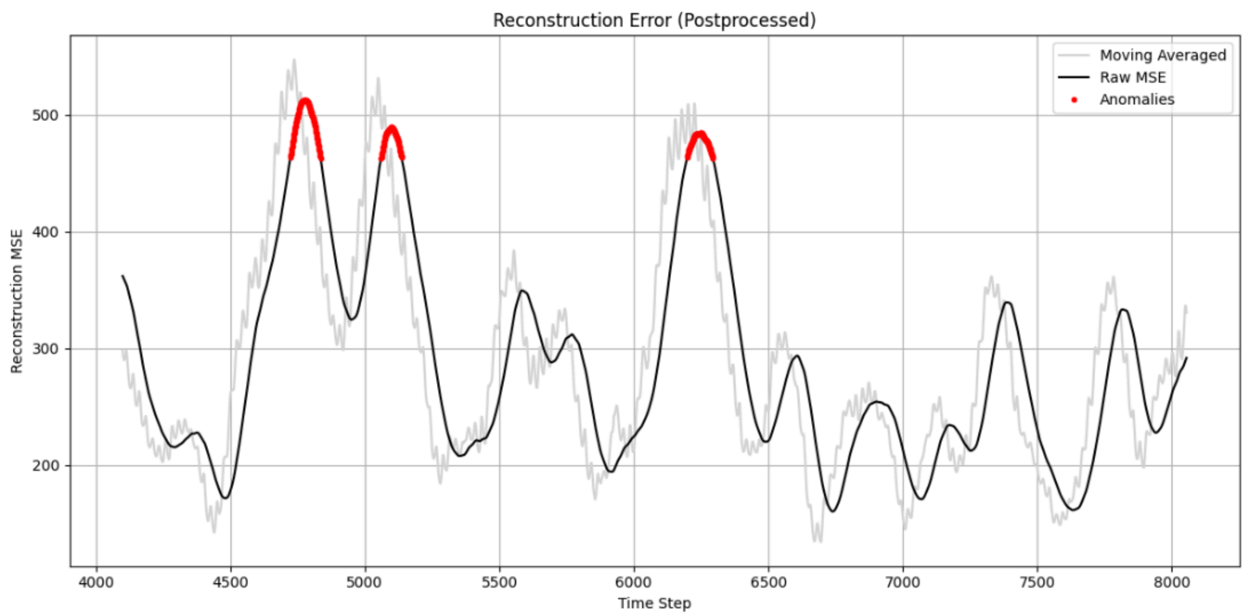




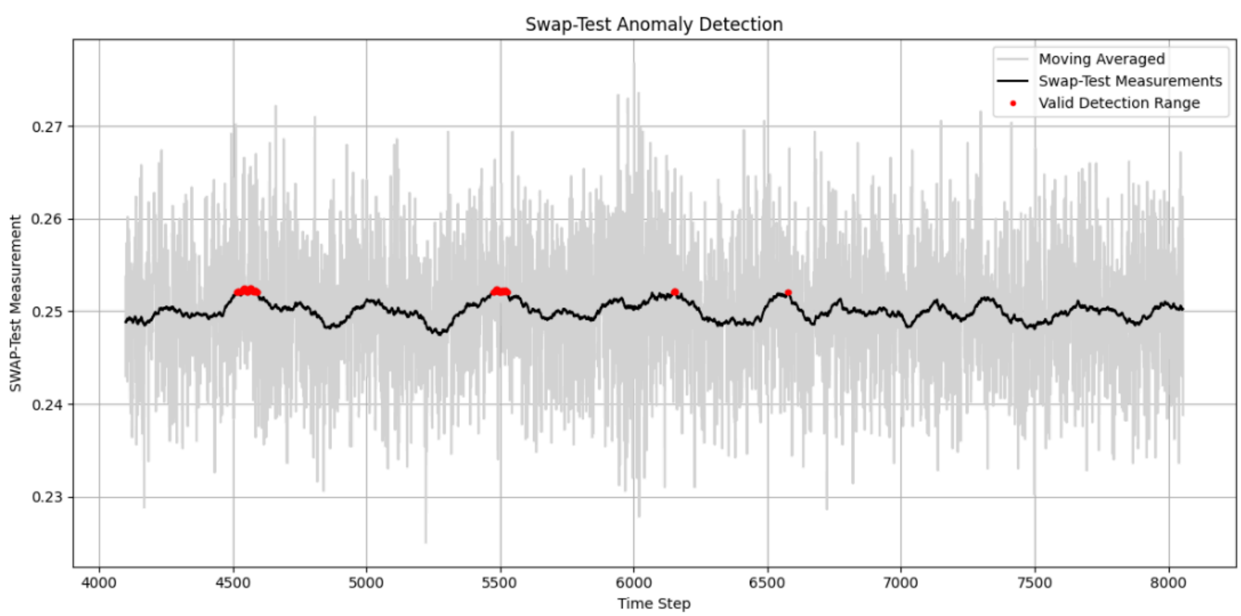
RealAmplitudes:

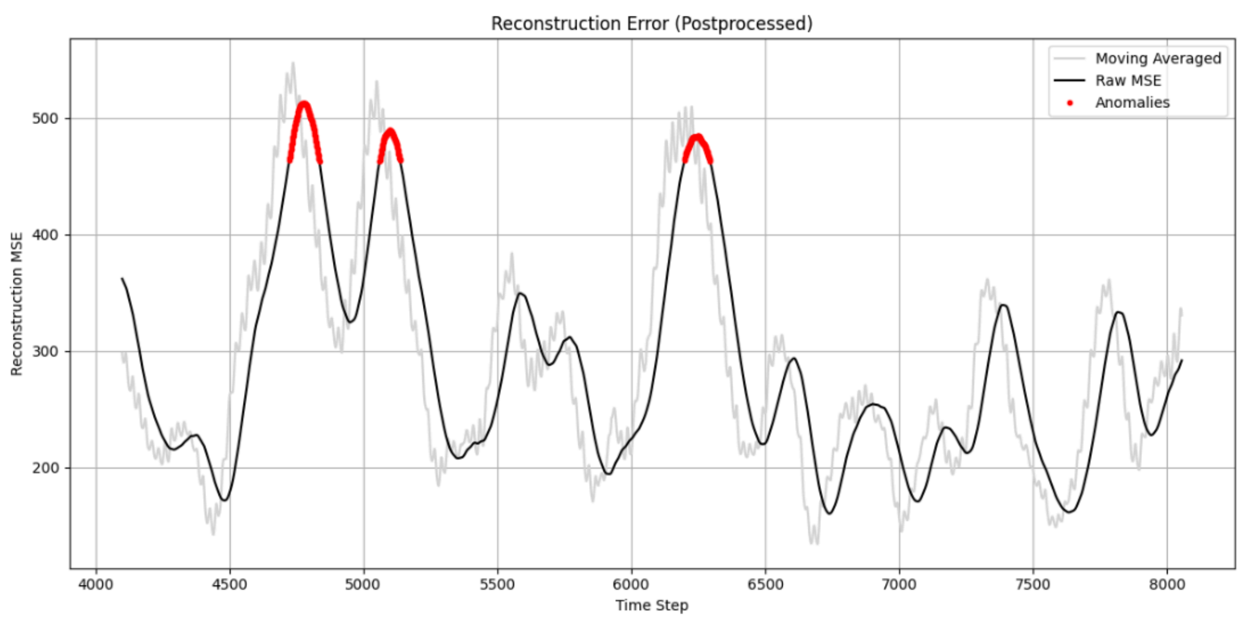
entanglement="circular"



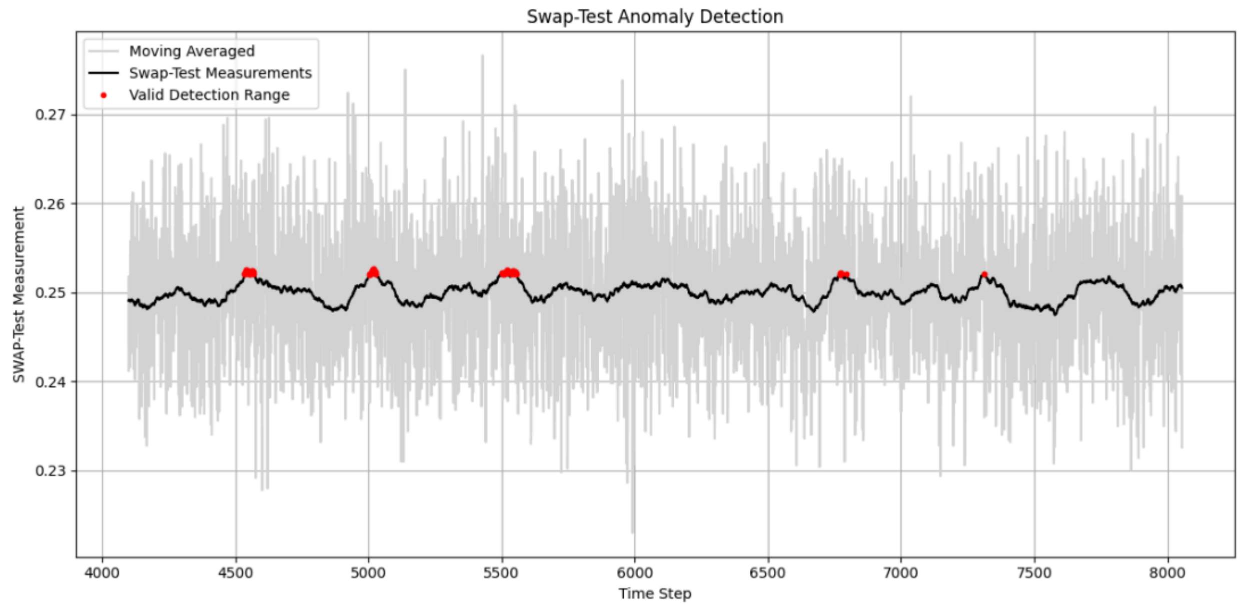


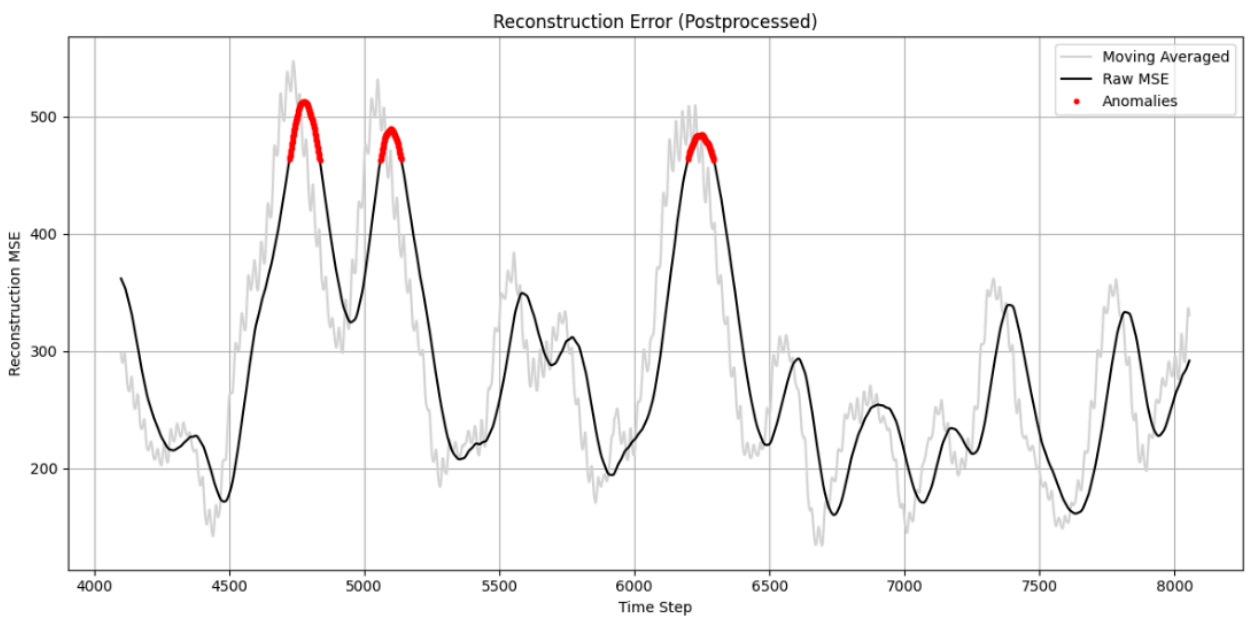
entanglement="full"



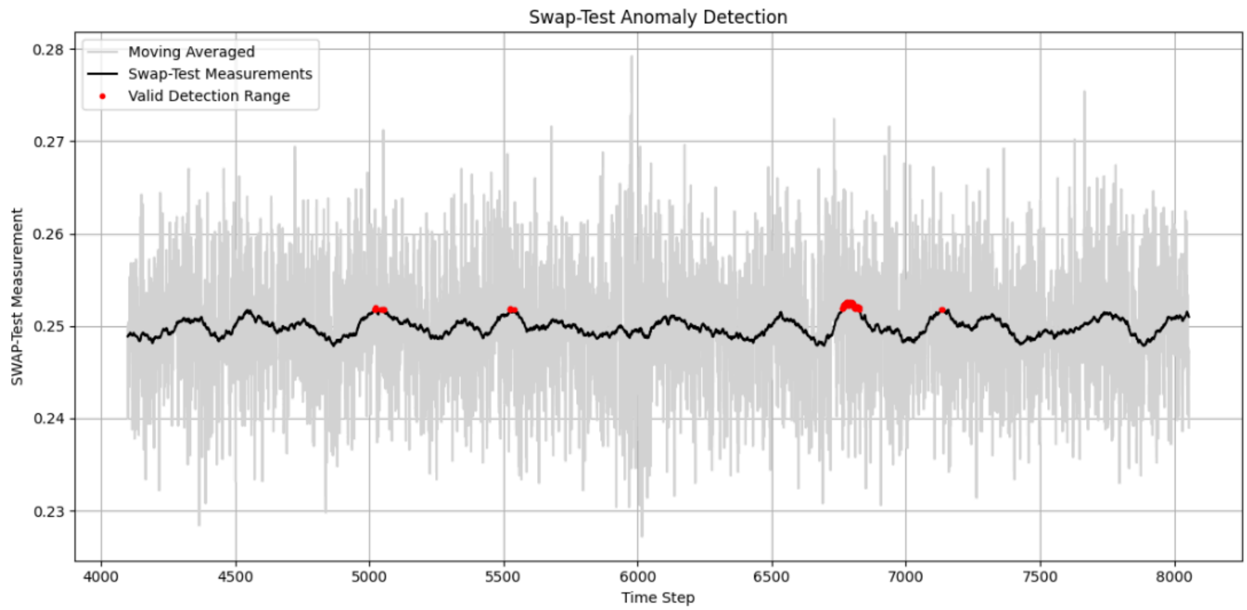


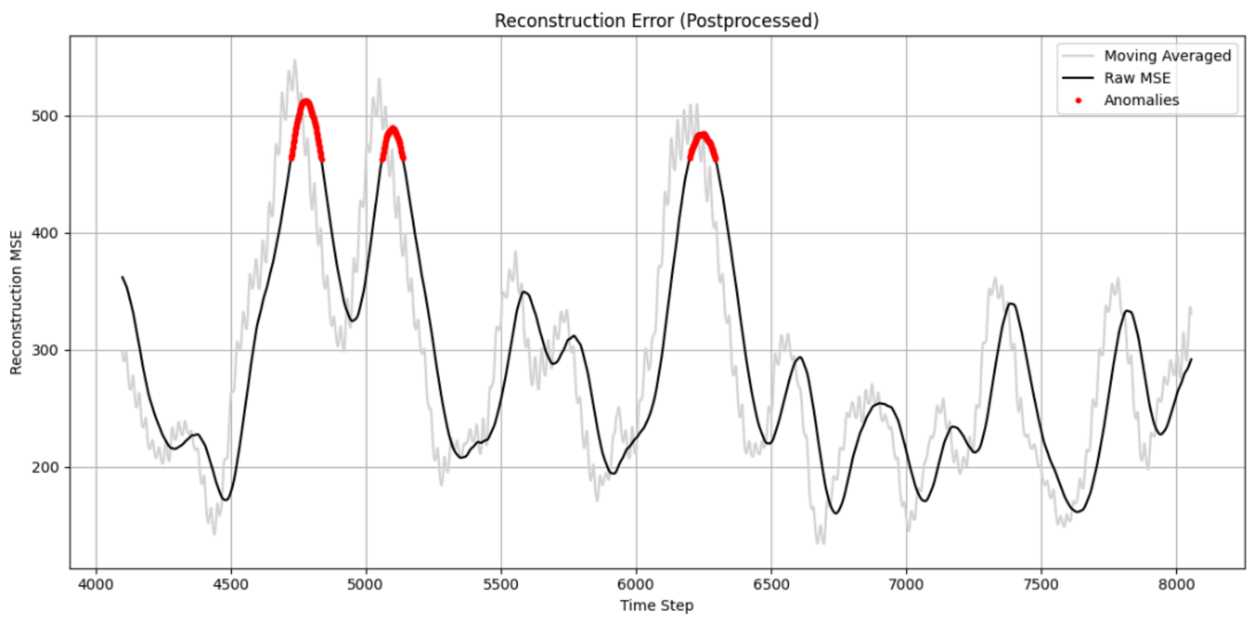
entanglement="linear"





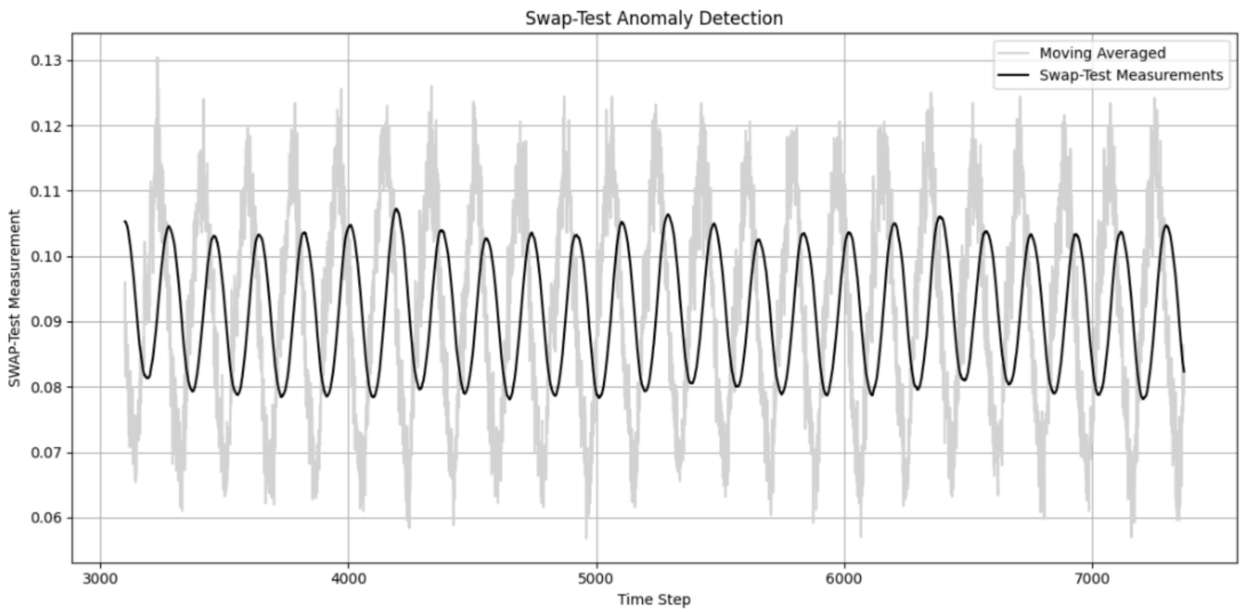
entanglement=”sca”

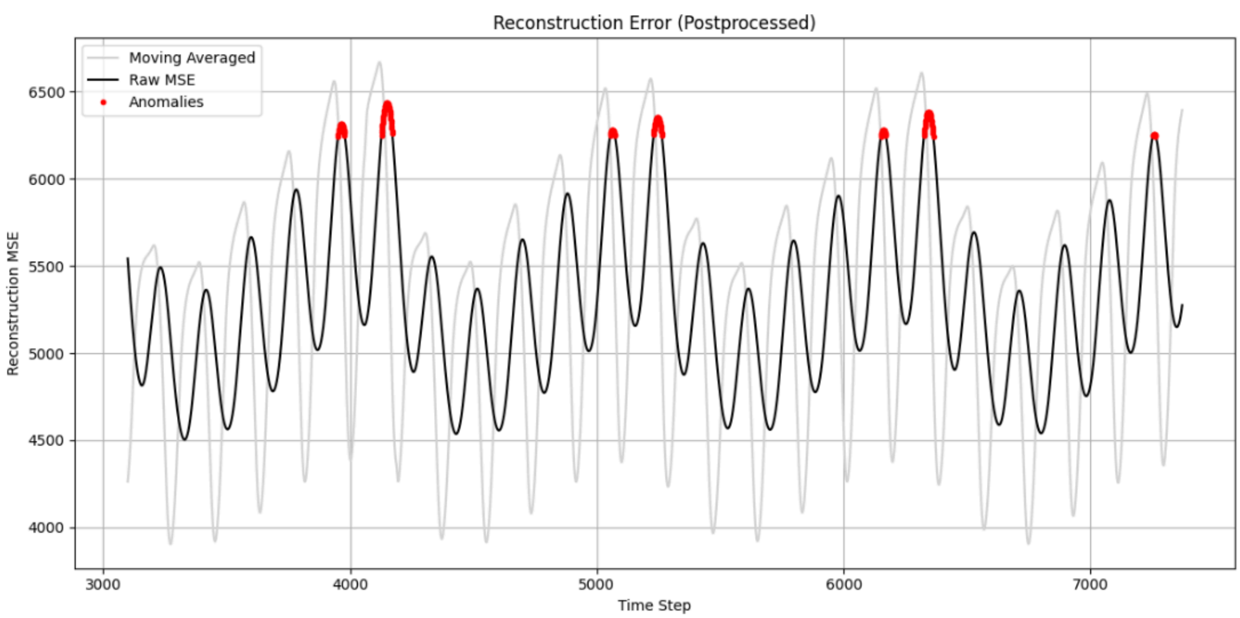




Dataset 138: 138_UCR_Anomaly_InternalBleeding19_3000_4187_4197.txt

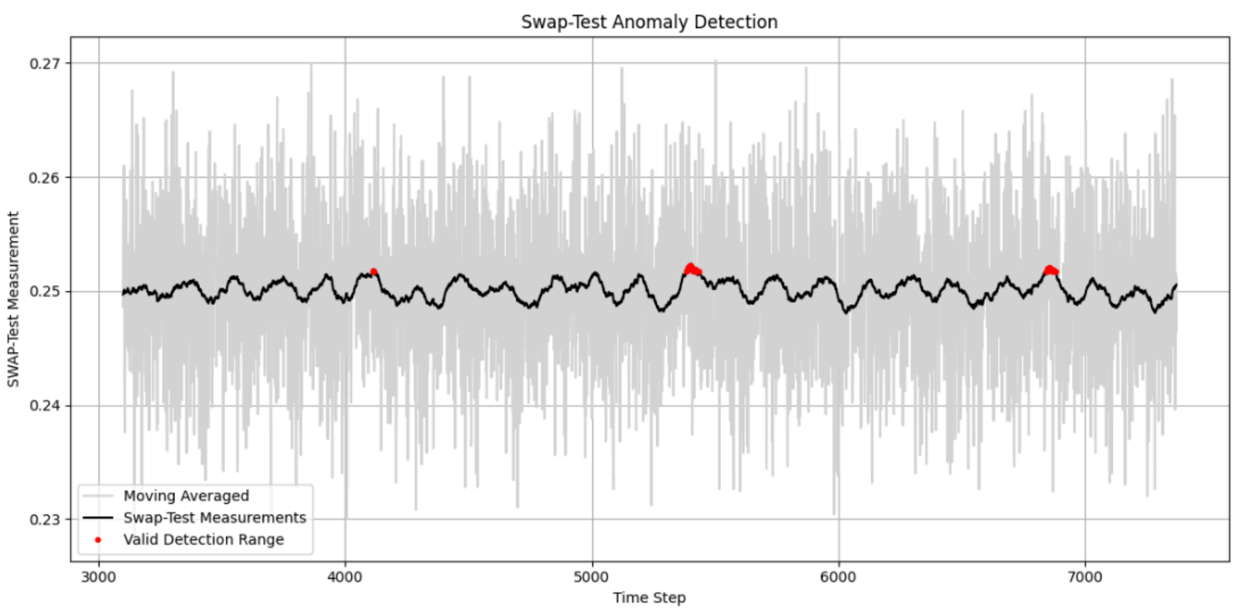
PauliTwoDesign:

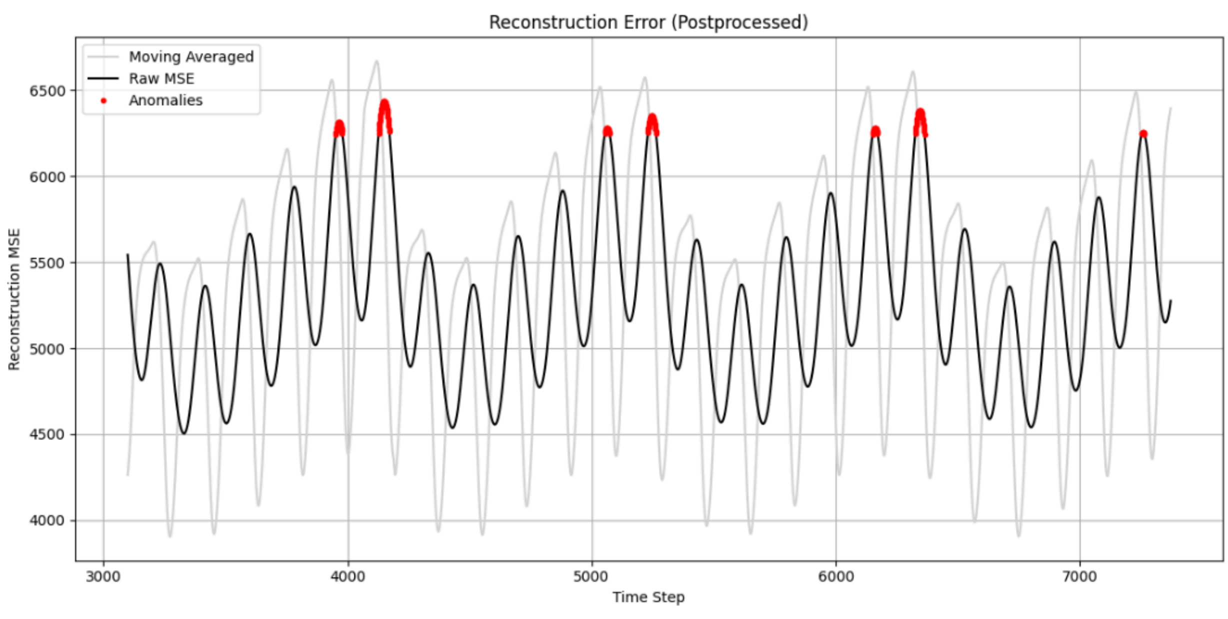




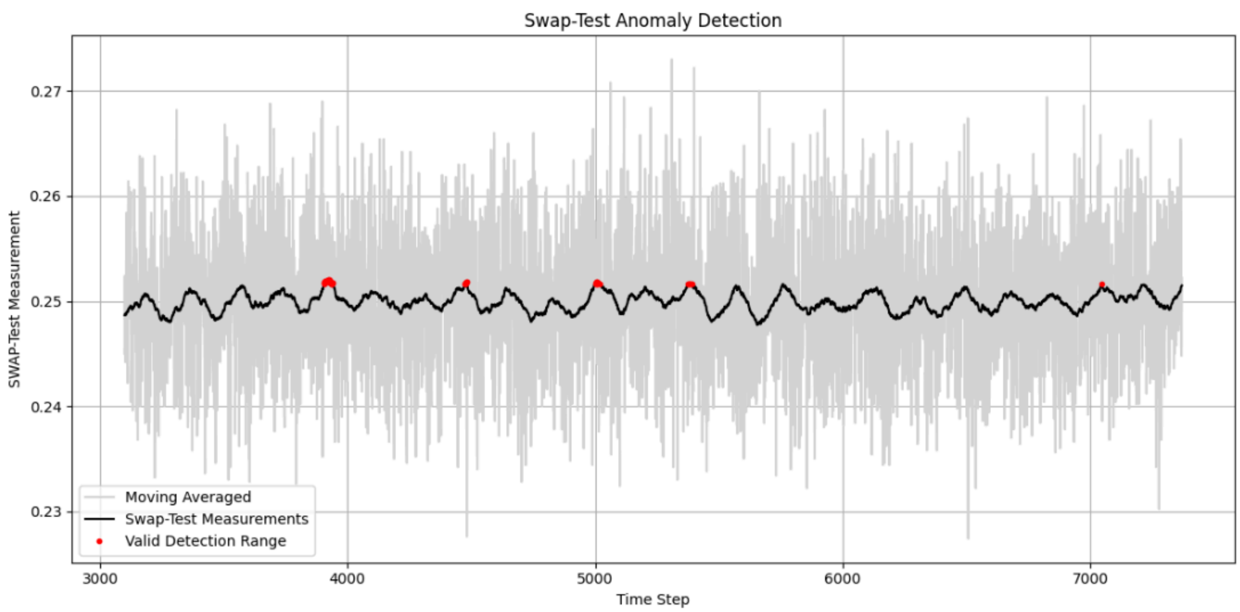
RealAmplitudes:

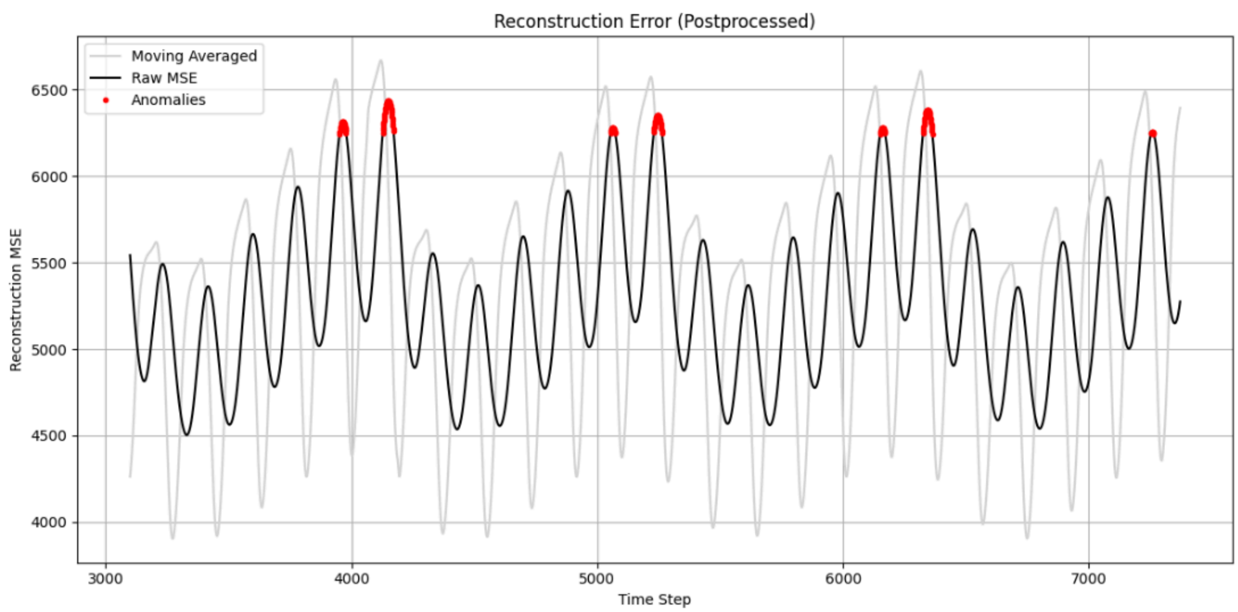
entanglement="circular"



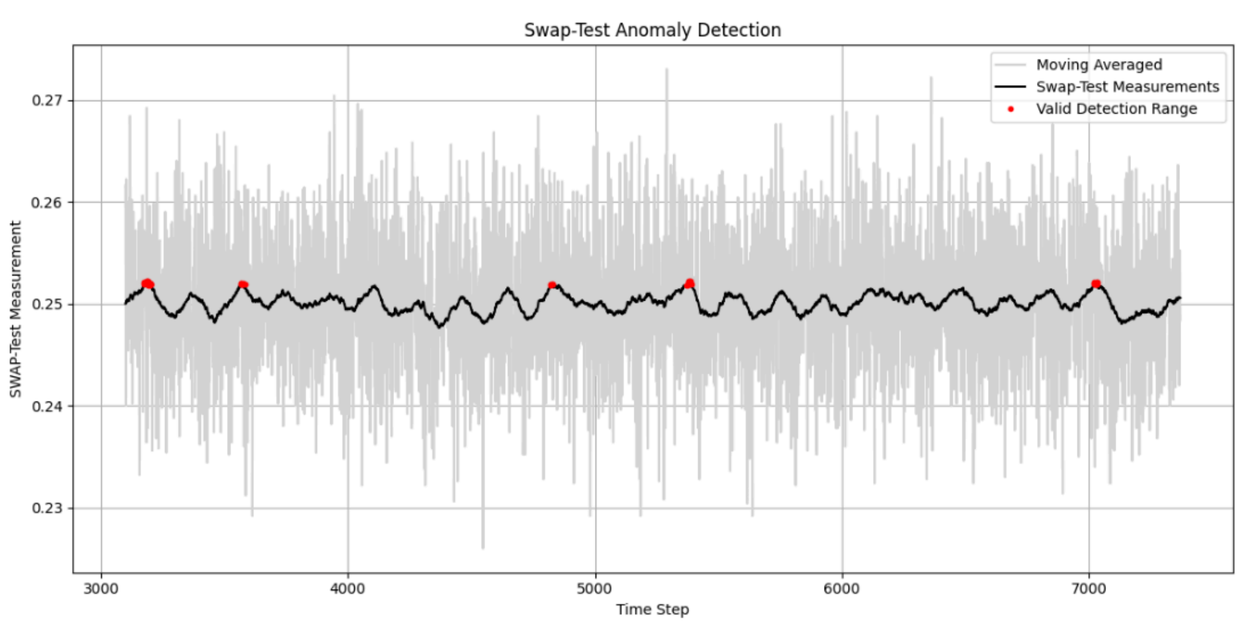


entanglement="full"

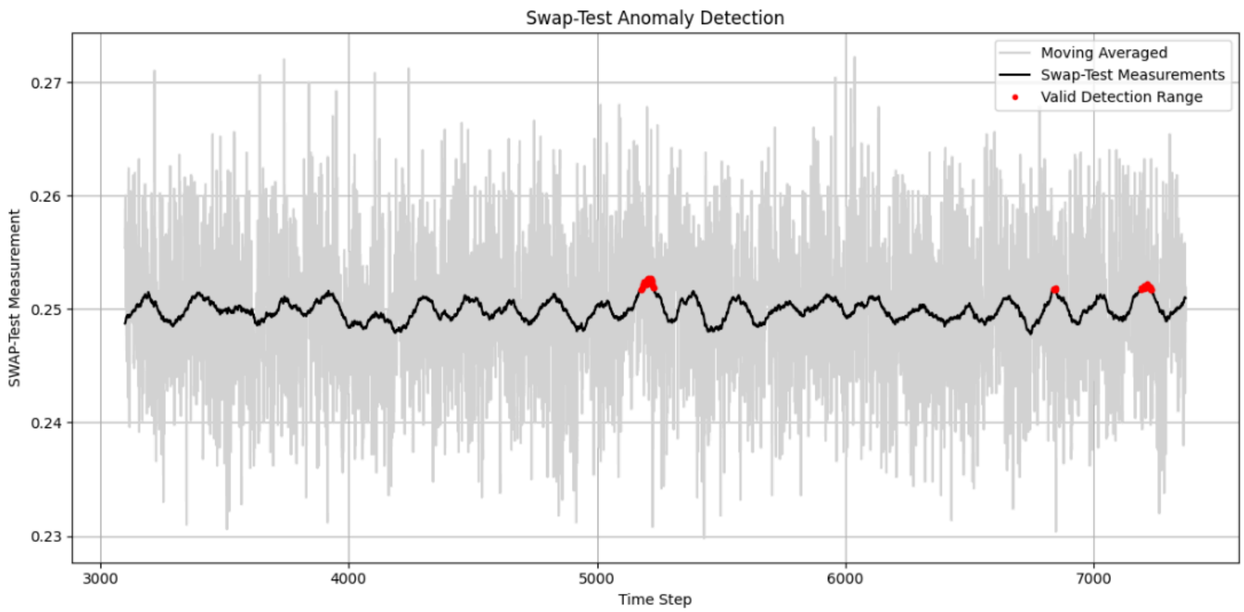


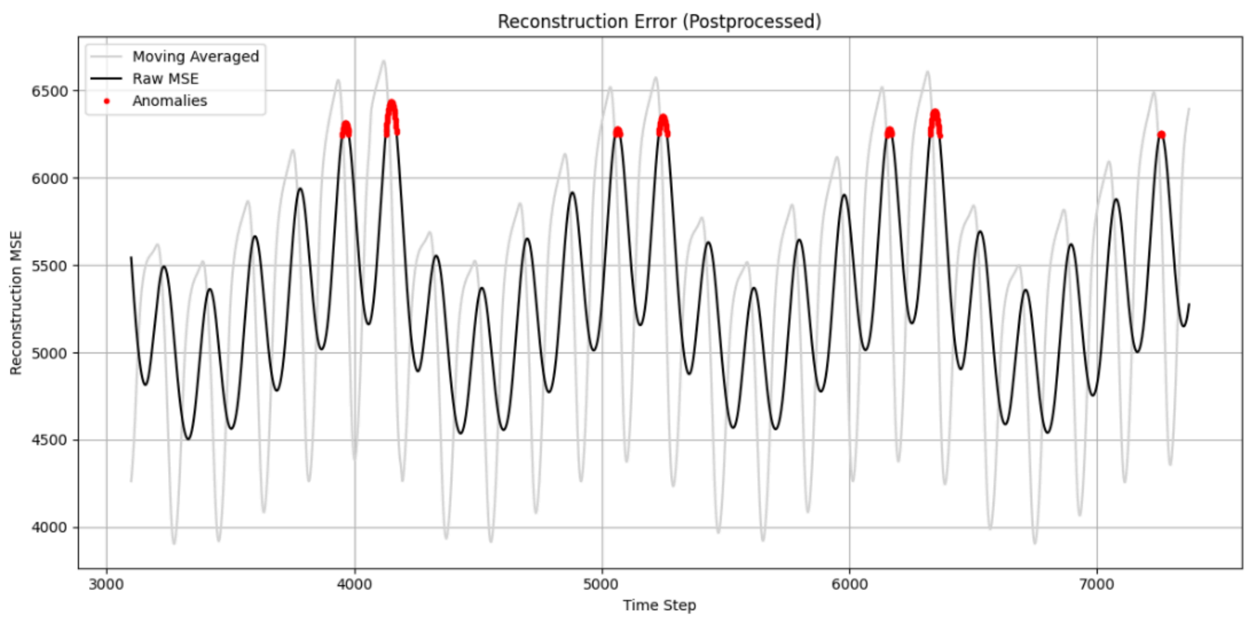


entanglement="linear"



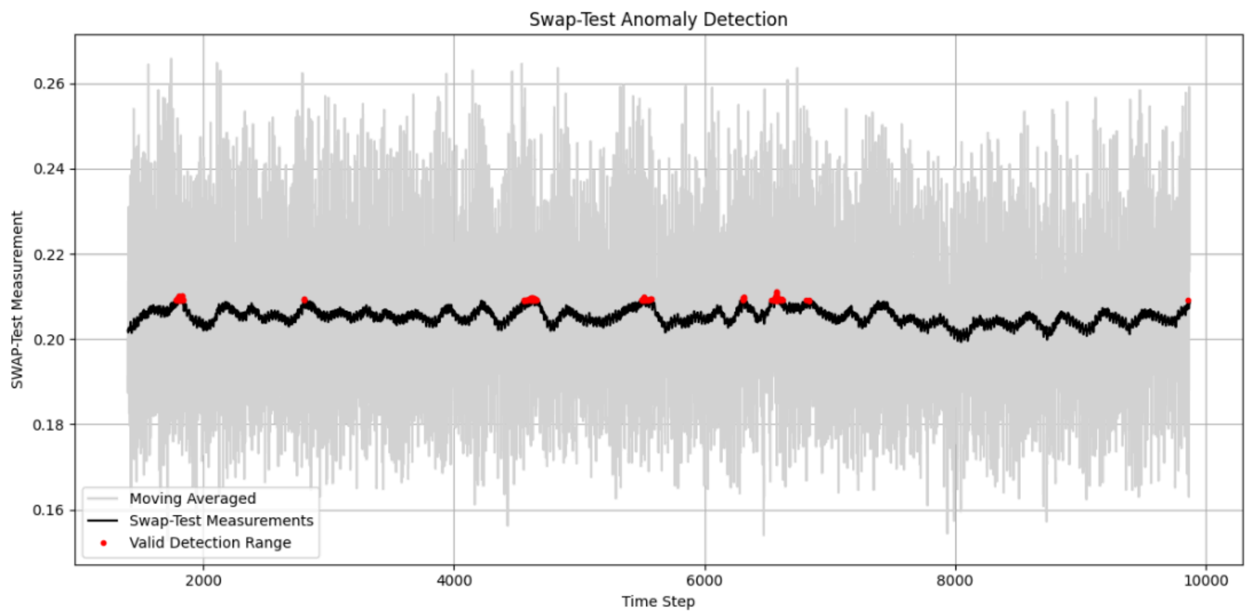
entanglement="sca"

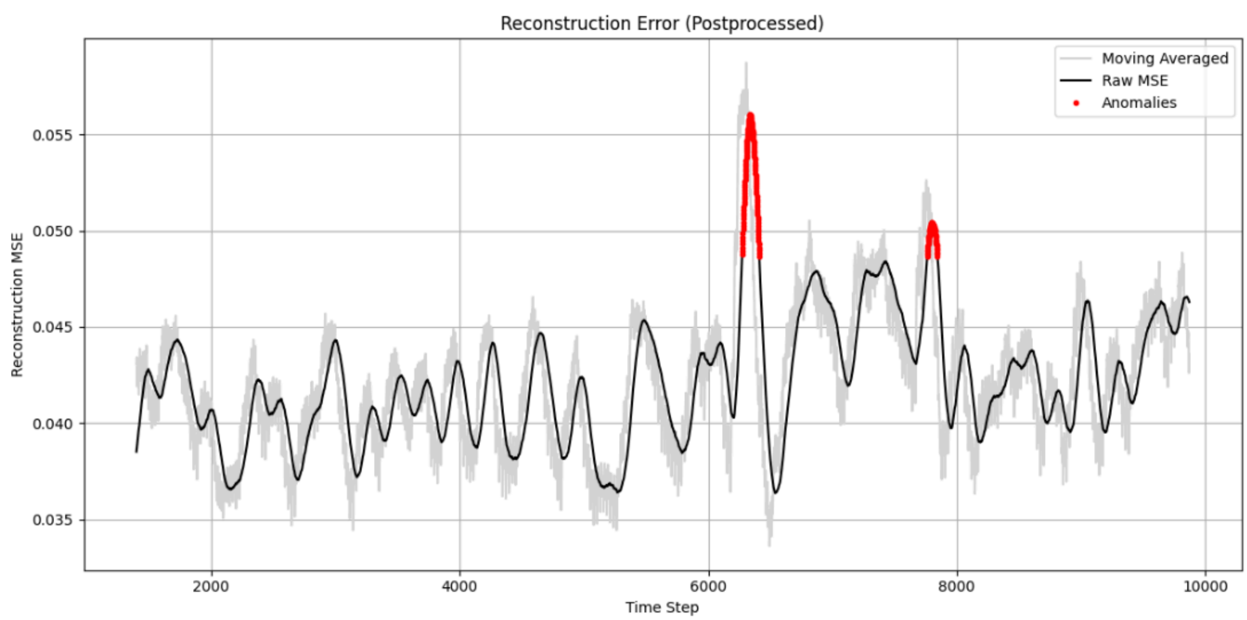




Dataset 176: 176_UCR_Anomaly_insectEPG4_1300_6508_6558.txt

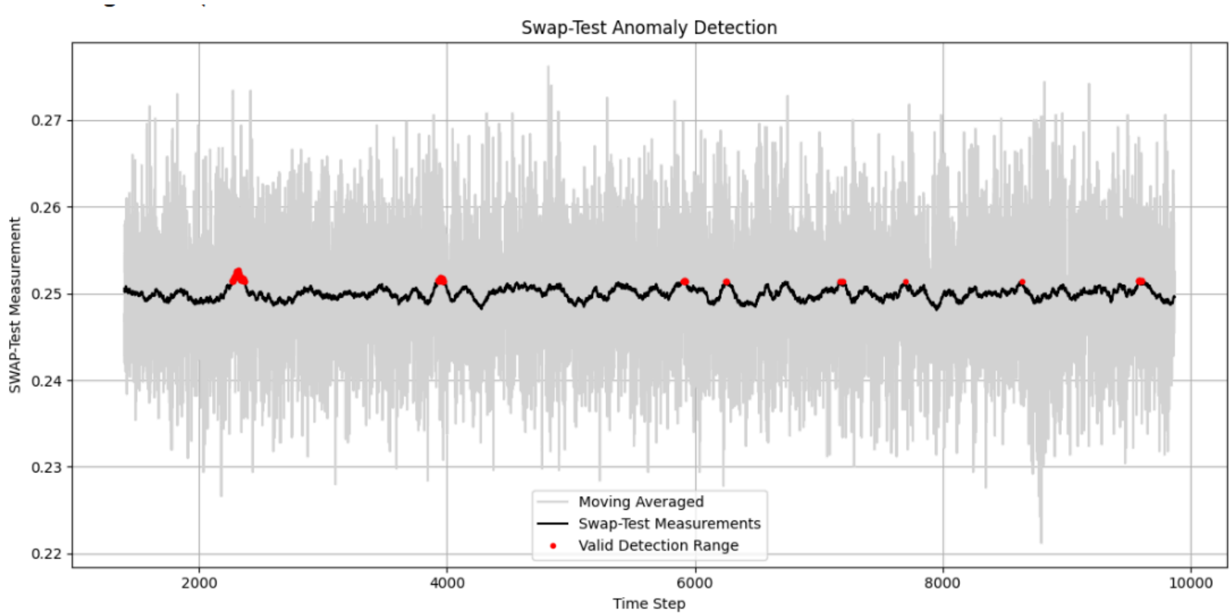
PauliTwoDesign:

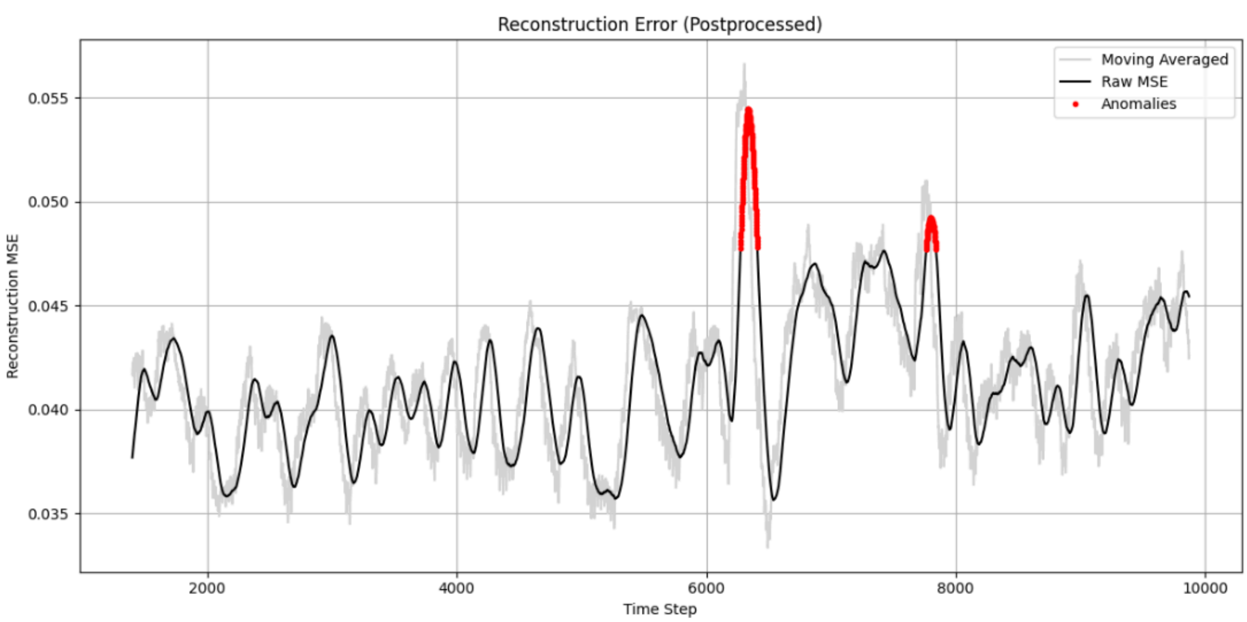




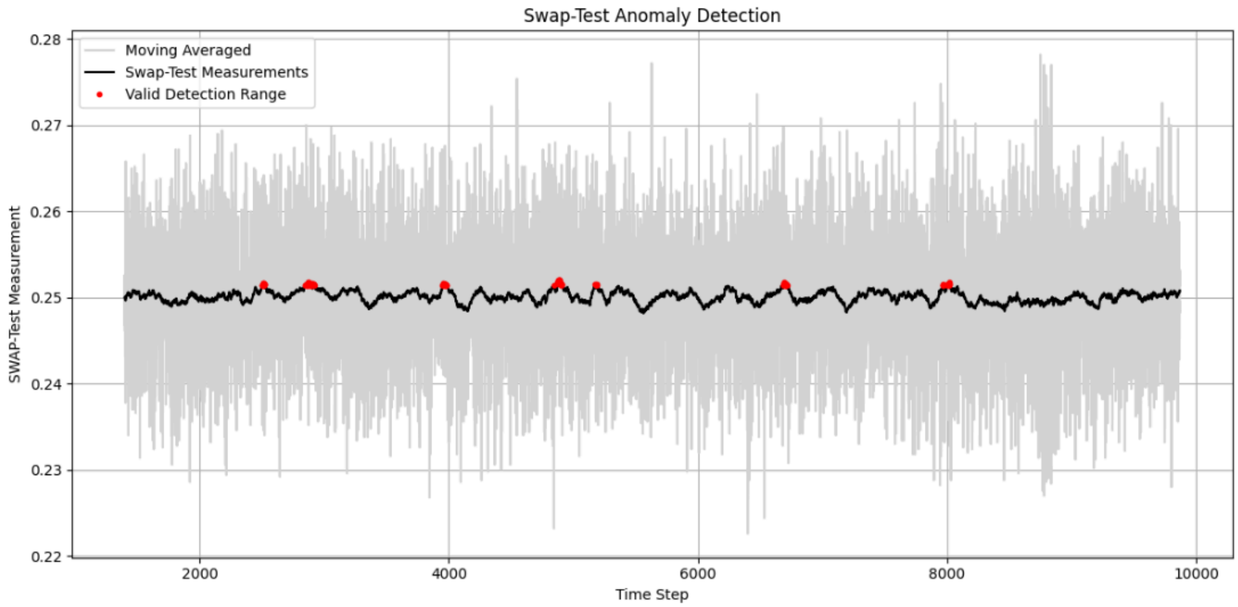
RealAmplitudes:

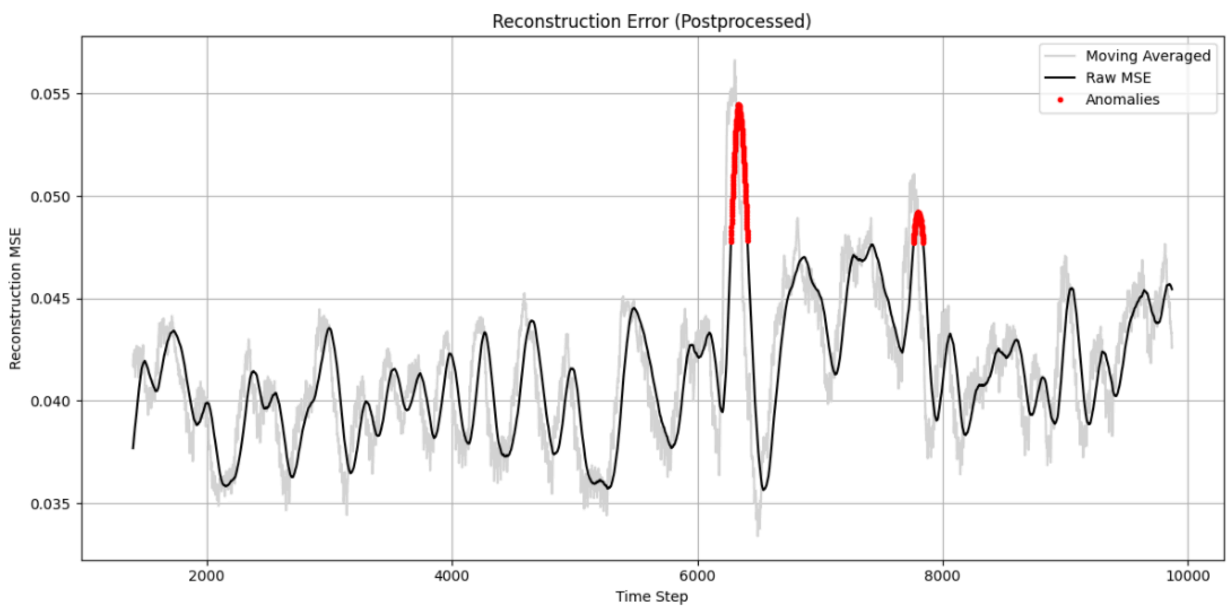
entanglement=”circular”



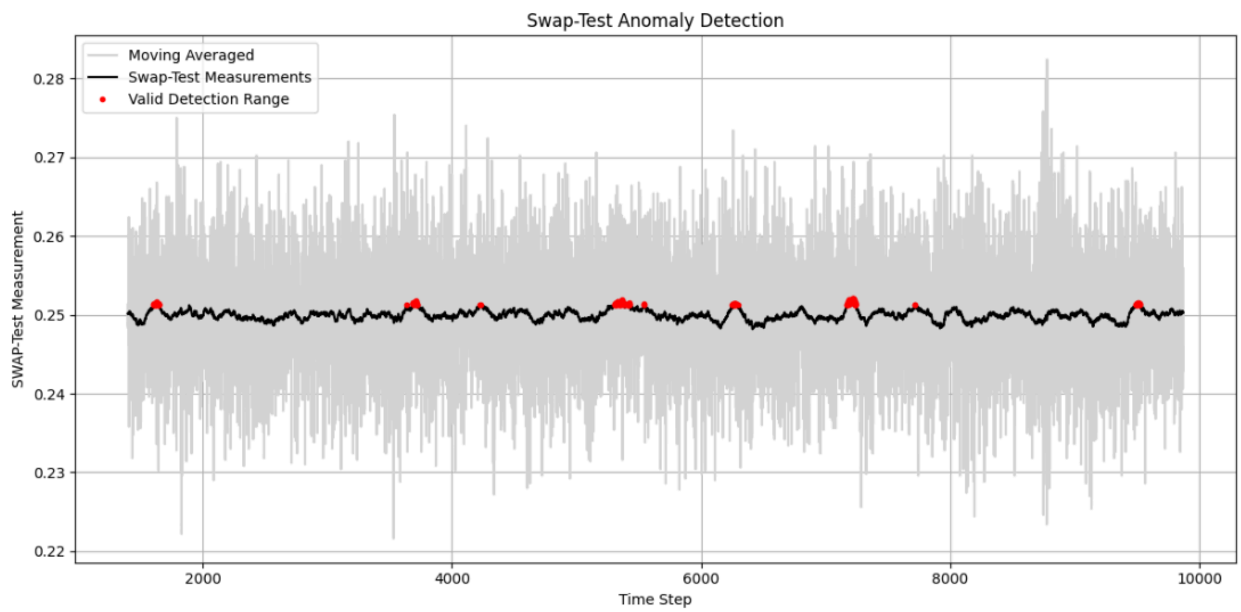


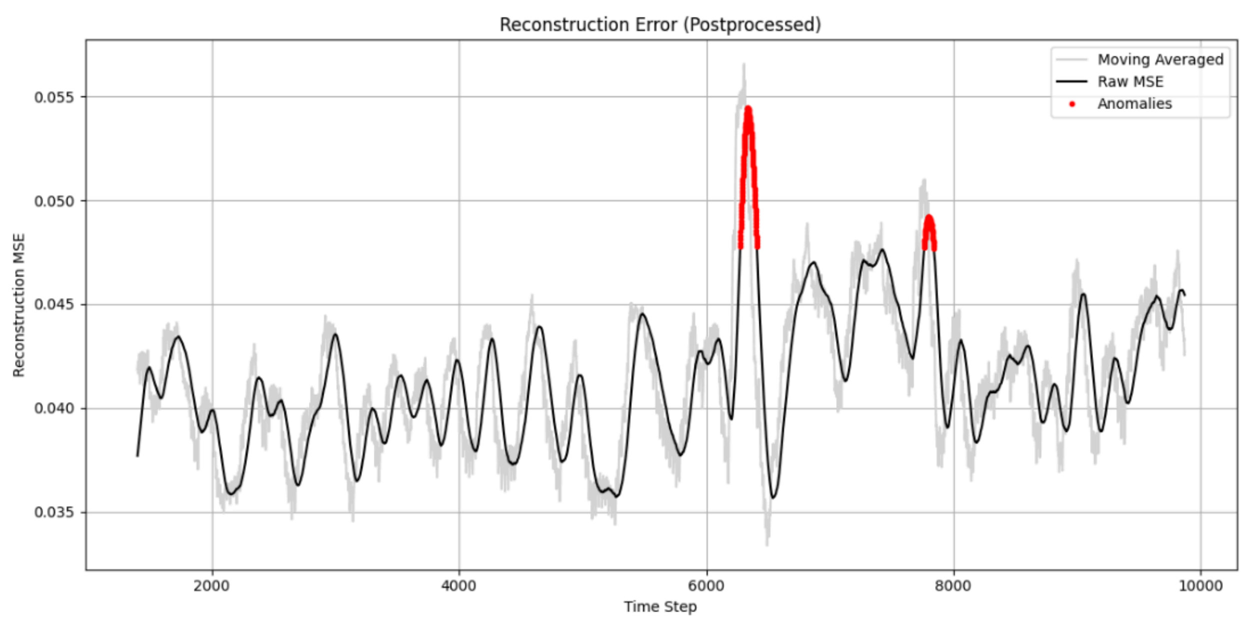
entanglement=”full”



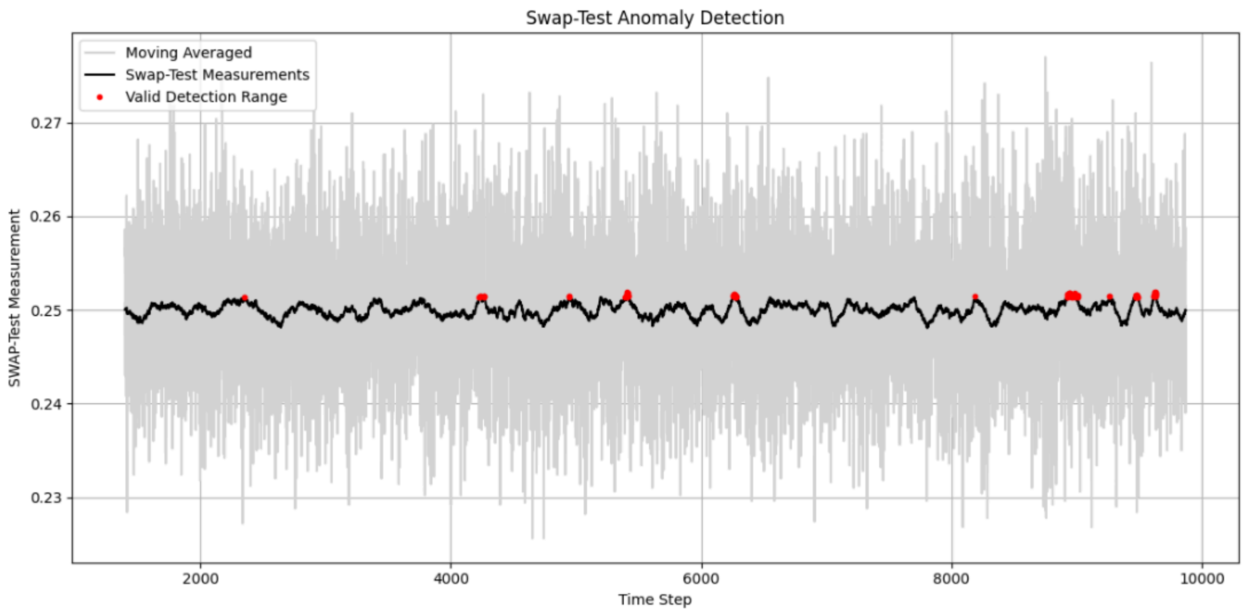


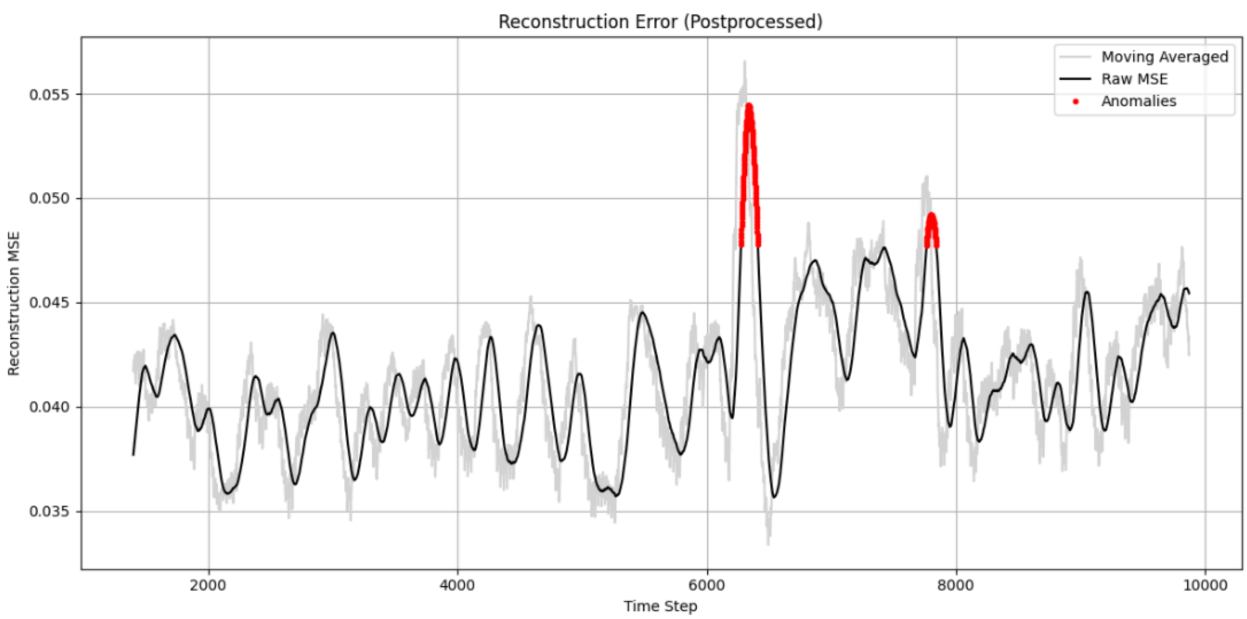
entanglement="linear"





entanglement="sca"





References

1. https://ieeexplore.ieee.org/document/8616075
2. https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.PauliTwoDesign
3. https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.RealAmplitudes
4. https://arxiv.org/pdf/2410.04154v2