QuAIL Tools for Benchmarking, Analysis and Quantum Algorithm Development

P.A. Lott, S. Mandra, J. Marshall, L. Brady, D. Bernal, A.A. Asanjan, S. Grabbe, A.A. Armenakas, M. Memarzadeh, T. Templin, E.G. Rieffel, R. Biswas

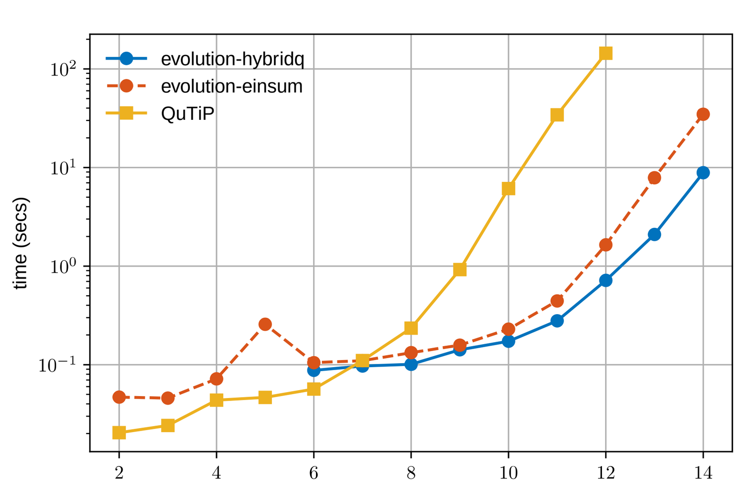
HybridQ and **PySA** are open-source tools developed by the *NASA Quantum Artificial Intelligence Laboratory* (QuAIL) to support benchmarking, analysis and quantum algorithm development in areas such as simulation, optimization and machine learning. These tools leverage classical hardware acceleration via high-performance computing CPU and GPU architectures and support high-performance computing. HybridQ is a highly extensible platform designed to provide a common framework to integrate multiple state-of-the-art techniques to simulate large-scale quantum circuits. PySA is an extensible platform to optimize a classical cost function. We provide an outline of these open-source tools and highlight projects using each of these tools in contexts of simulation, optimization and machine learning.

HYBRIDQ is a highly extensible platform designed to provide a common framework to integrate multiple state-of-the-art techniques to simulate large scale quantum circuits on a variety of hardware.

- Provides tools to manipulate, develop, and extend noiseless and noisy circuits for different hardware architectures.
- Supports large-scale high-performance computing (hpc) simulations, automatically balancing workload among different processor nodes and enabling the use of multiple backends to maximize parallel efficiency.
- Simple and expressive language allows for seamless switching from one technique to another as well as from one hardware to the next.
- Greatly simplifies the development of new hybrid algorithms and techniques.

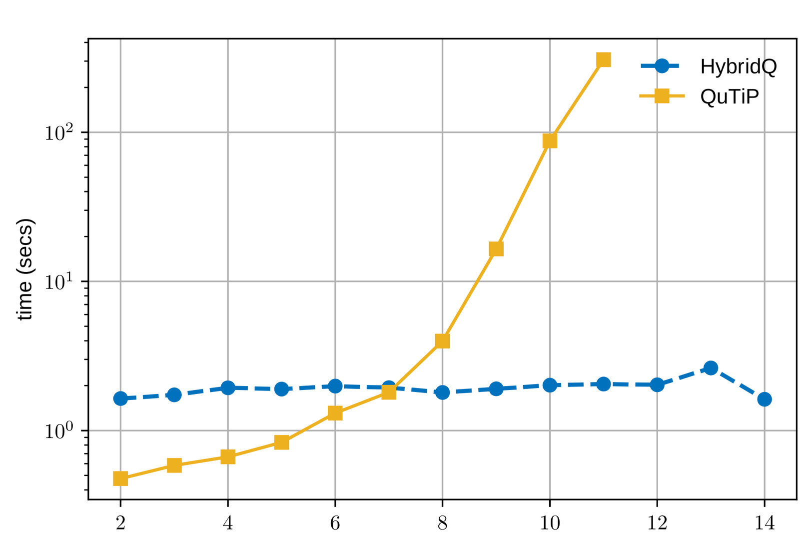
HTTPS://GITHUB.COM/NASA/HYBRIDQ

S. Mandra, J. Marshall, E.G. Rieffel, and R. Biswas, **HybridQ: A Hybrid Simulator for Quantum Circuits,** *IEEE/ACM Second International Workshop on Quantum Computing Software (QCS) 2021.*

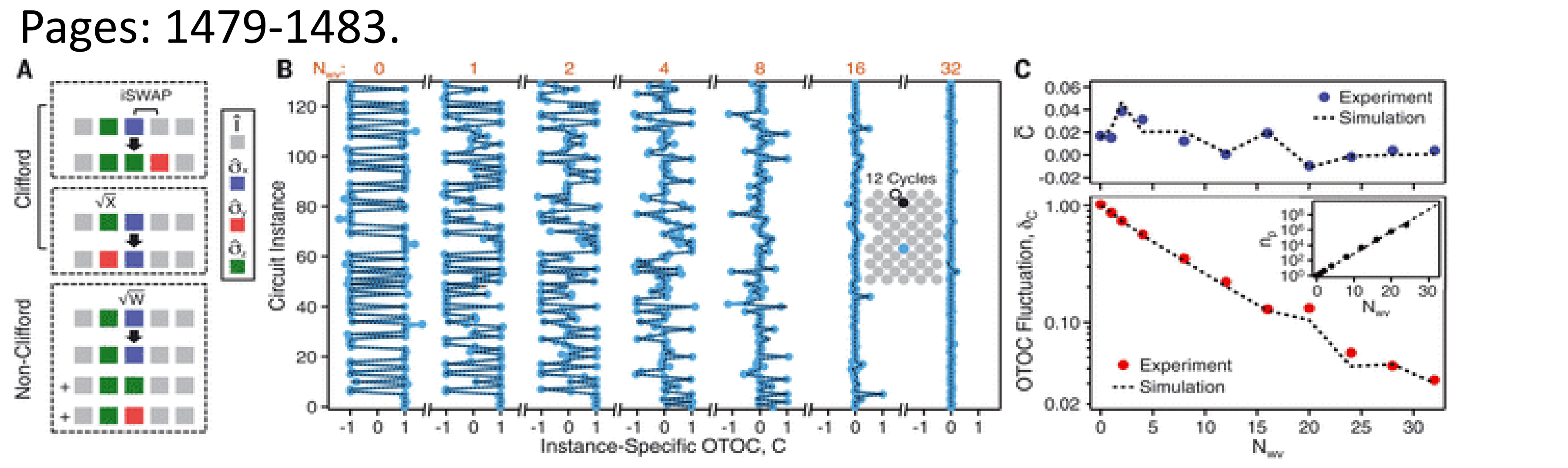


(left) Comparison of QuTiP to **HybridQ** speed on density matrix operations. Wall-clock time versus system size n (number of qubits). Depolarizing channel applied on each qubit of a random density matrix, utilizing two of the **HybridQ** backends. Note the 'evolution-hybridq' backend only works for more than 5 qubits in the density matrix setting.

(right) Tensor network contraction on a density matrix circuit. For each system size, we generate a random quantum circuit of 50 total gates, which we then add depolarizing noise after each gate. Using the tensor network contraction, we extract the reduced state of the 0'th qubit. In QuTiP we obtain this state by performing the full simulation and tracing out the other degrees of freedom.



Xiao MI et al. (Google team & NASA QuAIL contributors) **Information scrambling in quantum circuits,** Science, Volume: 374, Issue: 6574, Pages: 1479-1483.



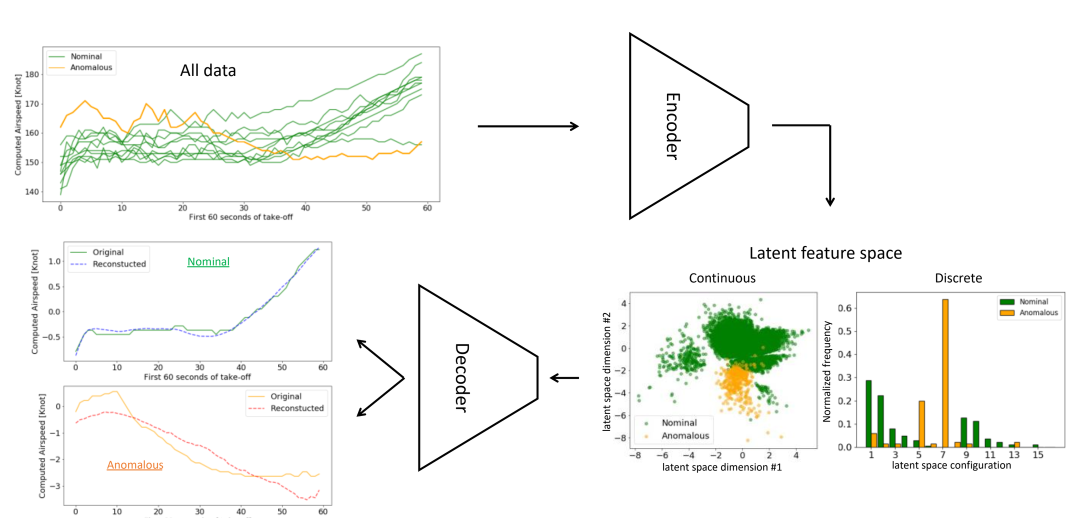
(**A**) Example transformation of a product of Pauli operators (Pauli string) by different quantum gates. A single Pauli string $\hat{I}(1)\hat{\sigma}(2)z\hat{\sigma}(3)x\hat{I}(4)\hat{I}(5)$ is mapped either into a different Pauli string by a Clifford gate or a superposition of multiple Pauli strings (coefficients not shown) by a non-Clifford gate. (**B**) OTOCs of individual random circuit instances, C, measured with the number of non-Clifford gates in $\hat{U}$, N_{wv}, fixed at different values. Dashed lines are numerical simulation results. For each circuit, the non-Clifford gates are injected at random locations within the intersection between the light cones of Q_b and Q_1. The inset shows locations of Q_a (black-outlined unfilled circle), Q_1 (black filled circle), and Q_b (blue filled circle) as well as the number of circuit cycles with which the data are taken. Here, and also in Fig. 4, error bars are omitted because a sufficient number of samples was taken to ensure that the statistical uncertainty is ≤0.01 (36). (**C**) The mean $\overline{C}$ (top) and RMS values δ_C (bottom) of C for different N_{wv}. Dashed lines are computed from the numerically simulated values in (B). (Inset) Numerically computed average numbers of Pauli strings in the time-evolved operator $\hat{O}(t)$, n_p, for the experimental circuits. Dashed line is an exponential fit, $n_p \approx 20.96 N_{wv}$. **HybridQ used to simulate 53 qubits with 32 non-Clifford gates.**

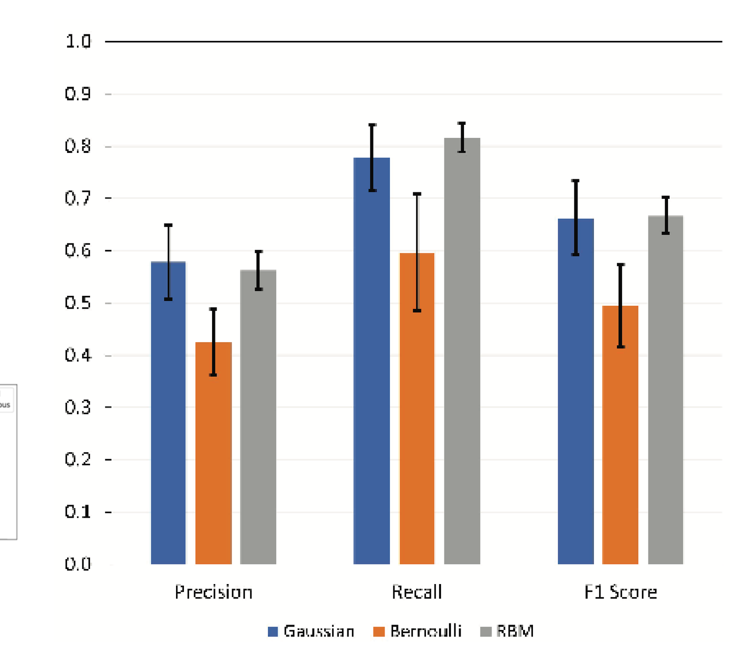
PYSA is an extensible platform to optimize classical cost functions.

- A powerful tool for solving optimization and sampling problems
- Enables use of different numerical techniques – including e.g. parallel-tempering, annealed importance sampling and path-integral monte carlo.
- Designed to run on HPC clusters via user-friendly interface
- Jupyter notebook tutorials available outlining use in solving ising models, and training restricted Boltzmann machines.

HTTPS://GITHUB.COM/NASA/PYSA

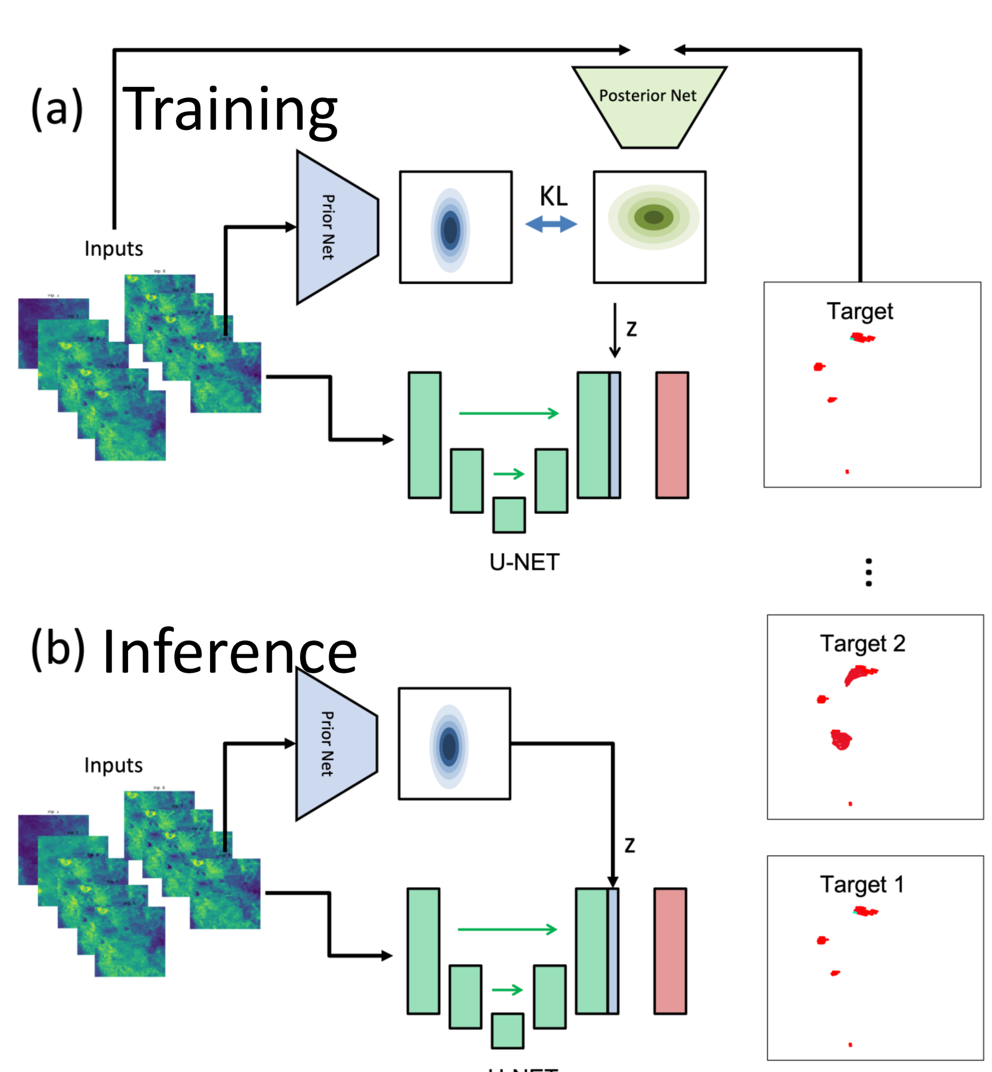
T. Templin, M. Memarzadeh, W. Vinci, P. A. Lott, A.A. Asanjan, A.A. Armenakas, E. G. Rieffel **Anomaly Detection in Aeronautics Data with Quantum-compatible Discrete Deep Generative Model,** *arXiv:2303.12302, 2023*





(left) unsupervised variational autoencoder model with convolutional encoder/decoder networks and latent prior for detecting anomalies in aeronautics time-series data. (right) Comparison of precision, recall and f1- scores for 3 different priors: Blue (Gaussian), Orange (Bernoulli), and Grey (Boltzmann) distributions, Boltzmann machine is comparable in performance to Gaussian prior, both outperforming Bernoulli.

A.A. Asanjan, , M. Memarzadeh, P. A. Lott, A.A. Asanjan, E. G. Rieffel, S. Grabbe **Probabilistic Wildfire Segmentation Using Deep Learning from Satellite Imagery,** *Remote Sensing 2023*



Probablistic U-Net to build a supervised stochastic segmentation via RBM latent model & feature learning. Method combines Variational Inference & MCMC to generate more accurate latent samples and provide realistic scenarios for wildfire detection.

Figure provides a graphical illustration of the proposed Probabilistic U-Net framework. The inputs are NDVI, NDVI difference with long-term NDVI, and MODIS MCD43A4 channels for Land/Cloud/Aerosols.

(a) (top) presents the training scheme where the prior network encodes inputs and the posterior network encodes the inputs and target data together into multivariate Gaussian distributions. The samples from the unified multivariate Gaussian distribution are concatenated with U-Net outputs to produce stochastic events of target data.

(b) (bottom) demonstrates the inference scheme where samples are drawn from the prior network.