

# qlass: A Python Package for Variational Quantum Algorithms on Photonic Devices

Farrokh Labib<sup>1,2</sup>, Akilan Rajamani<sup>3</sup>, Bruno Senjean<sup>3</sup>, and Nathan Shammah<sup>1,2</sup>

<sup>1</sup> Unitary Fund France, Tours, France <sup>2</sup> Unitary Foundation, San Francisco, USA <sup>3</sup> ICGM, Univ Montpellier, CNRS, ENSCM, Montpellier, France

DOI: [10.21105/joss.09316](https://doi.org/10.21105/joss.09316)

## Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Daniel S. Katz](#)

## Reviewers:

- [@goerz](#)
- [@burgholzer](#)

Submitted: 25 September 2025

Published: 15 January 2026

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

qlass is a Python package designed to enable the execution of variational quantum algorithms on photonic quantum computing devices. Part of the Quantum Glass-based Photonic Integrated Circuits (QLASS) project funded by the European Union, this package bridges the gap between quantum algorithm development in popular frameworks like Qiskit and their implementation on photonic hardware using Perceval.

## Statement of need

Photonic quantum computing represents a powerful approach to building scalable quantum computers, offering advantages such as avoiding the need for dilution fridges, providing natural connectivity to quantum communication networks (Kok et al., 2007; O'Brien et al., 2009) and, as in the case of QLASS devices, allowing for reconfigurable chips. However, the translation of quantum algorithms from the abstract circuit model to photonic implementations presents unique challenges. Existing quantum software frameworks like Qiskit (Aleksandrowicz et al., 2019) and Cirq (Cirq Developers, 2025) primarily target gate-based quantum computers with qubit architectures, while photonic platforms operate on fundamentally different principles using linear optical elements, photons, and modes.

There are only a few open source software projects that focus on the simulation or compilation of photonic quantum computers. These include Perceval (Heurtel et al., 2023), Strawberry Fields (Killoran et al., 2019), Piquasso (Kolarovszki et al., 2025), and Graphix (Sunami & Fukushima, 2022), all Python packages. These tools are primarily focussed on photonic circuit compilation and simulation. We are, however, interested in specific applications of photonic quantum computing, in particular, in applications related to quantum chemistry. As far as we know, there is no software platform that can run VQE simulations on photonic quantum computing platforms starting with an ansatz defined in the qubit architecture all the way to running the VQE on a photonic quantum computer to obtain ground state energies for molecular Hamiltonians.

qlass addresses this gap by providing:

1. **Quantum chemistry integration** with established packages (OpenFermion, PySCF) for molecular problem definition
2. **Optimized VQE implementations** specifically designed for the constraints and capabilities of photonic quantum computers
3. **Hardware-specific resource estimation and compilation** from quantum circuits (Qiskit) to linear optics layout of photonic processors (Perceval), including the analysis of circuits against realistic hardware constraints such as photon loss, detector efficiency, and fusion

gate success rates

The package is particularly valuable for researchers working at the intersection of quantum algorithms and photonic quantum computing, as it provides the necessary tools to evaluate algorithm performance under realistic hardware conditions and optimize implementations for photonic architectures.

## Package Architecture and Features

qlass provides an end-to-end pipeline for quantum chemistry simulations on photonic quantum computers:

```
from qlass.quantum_chemistry import LiH_hamiltonian
from qlass.vqe import VQE, le_ansatz
from perceval.algorithm import Sampler

# Complete pipeline: molecule → photonic VQE → ground state energy
hamiltonian = LiH_hamiltonian(num_electrons=2, num_orbitals=1)

def executor(params, pauli_string):
    processor = le_ansatz(params, pauli_string) # Includes compilation step
    sampler = Sampler(processor)
    return sampler.samples(10_000)

vqe = VQE(hamiltonian=hamiltonian, executor=executor, num_params=4)
energy = vqe.run(max_iterations=10)
```

Key features include resource-aware compilation with realistic hardware modeling, multiple VQE executor types (sampling, qubit unitary, photonic unitary), automatic Pauli term grouping for measurement optimization, and specialized ansätze.

## Implementation Details

The package is structured into four main modules:

1. **compiler**: Handles circuit translation and resource analysis
2. **quantum\_chemistry**: Provides Hamiltonian generation and classical solutions
3. **vqe**: Implements variational algorithms and ansätze
4. **utils**: Contains utility functions for measurement processing and expectation value calculations

The implementation leverages several design patterns to ensure extensibility. It allows integration with different quantum backends and enables easy addition of new variational forms. We also support multiple output formats from different simulator backends.

## Distribution and Development

The qlass source code is hosted on Unitary Foundation's GitHub repository. It is released there and via the Python Package Index (PyPI). The documentation can be found on <https://qlass.readthedocs.org/> and consists of a user guide, an API doc, and tutorials. Milestones are used to guide development sprints and coordinate with contributors. It is licensed under the permissive Apache 2.0 licence to facilitate its adoption and integration in the developing quantum software stack.

## Usage, Contributions and Community

Users can open issues on the GitHub repository to receive assistance from maintainers or use a dedicated Discord channel on the Unitary Foundation Discord server for broader support and engagement. `qlass` is already used to support theoretical and experimental work in academia and in the quantum industry, such as in the QCLASS project, where it is used to model experiments with the glass-based processors by the startup Ephos and to test quantum chemistry solutions in Lithium ion battery modeling by CNRS and the University of Montpellier. `qlass` participated in unitaryHACK, the largest hackathon in quantum open-source software, where it was the first project, out of over 50 participating ones, to have all bounties awarded, gaining two new contributors: a student from the University of Boston and a developer from Mexico. `qlass` has been used as training material for the Collaborative Innovative Network by the European Space Agency's Phi Lab.

## Conclusions and Future Work

`qlass` provides essential infrastructure for implementing variational quantum algorithms on photonic quantum computers. By bridging the gap between abstract quantum circuits and photonic hardware implementations, it enables researchers to explore the potential of photonic platforms for quantum chemistry and optimization problems.

Future development priorities include:

- Integration with additional photonic simulation platforms
- Optimization algorithms specifically designed for photonic architectures
- Extended support for error mitigation techniques in the photonic context

The package is actively maintained by the Unitary Foundation as part of the QCLASS project, with contributions welcome from the quantum computing community.

## Acknowledgements

We acknowledge support from the European Union through the QCLASS project (EU Horizon Europe grant agreement 101135876). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. We thank the contributors to the open-source quantum computing ecosystem, particularly the developers of Qiskit, Perceval, OpenFermion, and PySCF. We would like to thank Jean-Sébastien Filhol and Francesco Malaspina for their valuable insight and the GitHub users Kitsunp and Qubit1718 for their code contributions.

## References

- Aleksandrowicz, G., Alexander, T., Barkoutsos, P., Bello, L., Ben-Haim, Y., Bucher, D., Cabrera-Hernández, F. J., Carballo-Franquis, J., Chen, A., Chen, C.-F., & others. (2019). Qiskit: An open-source framework for quantum computing. *Accessed on: Mar, 16, 61*. <https://doi.org/10.5281/zenodo.2562111>
- Cirq Developers. (2025). *Cirq*. Zenodo. <https://doi.org/10.5281/zenodo.4062499>
- Heurtel, N., Fyrrillas, A., De Glinasty, G., Le Bihan, R., Malherbe, S., Pailhas, M., Bertasi, E., Bourdoncle, B., Emeriau, P.-E., Mezher, R., & others. (2023). Perceval: A software platform for discrete variable photonic quantum computing. *Quantum*, 7, 931. <https://doi.org/10.22331/q-2023-02-21-931>

- Killoran, N., Izaac, J., Quesada, N., Bergholm, V., Amy, M., & Weedbrook, C. (2019). Strawberry fields: A software platform for photonic quantum computing. *Quantum*, 3, 129. <https://doi.org/10.22331/q-2019-03-11-129>
- Kok, P., Munro, W. J., Nemoto, K., Ralph, T. C., Dowling, J. P., & Milburn, G. J. (2007). Linear optical quantum computing with photonic qubits. *Reviews of Modern Physics*, 79(1), 135–174. <https://doi.org/10.1103/RevModPhys.79.135>
- Kolarovszki, Z., Rybotycki, T., Rakyta, P., Kaposi, Á., Poór, B., Jóczik, S., Nagy, D. T., Varga, H., El-Safty, K. H., Morse, G., & others. (2025). Piquasso: A photonic quantum computer simulation software platform. *Quantum*, 9, 1708. <https://doi.org/10.22331/q-2025-04-15-1708>
- O'Brien, J. L., Furusawa, A., & Vučković, J. (2009). Photonic quantum technologies. *Nature Photonics*, 3(12), 687–695. <https://doi.org/10.1038/nphoton.2009.229>
- Sunami, S., & Fukushima, M. (2022). Graphix: Optimizing and simulating measurement-based quantum computation on local-clifford decorated graph. *arXiv Preprint arXiv:2212.11975*. <https://doi.org/10.48550/arXiv.2212.11975>