ADAPT-VQE Lightning Pipeline: Inputs, Transformations, and Outputs

Thomas Roustan

Pierre Lapolla
pro@pierrelapolla.com

***Abstract*—This document presents the current ADAPT-VQE Lightning implementation as an explicit process specification. It focuses on three layers: the molecular and runtime inputs, the mathematical and computational transformations applied to those inputs, and the quantitative outputs produced by training and checkpoint artifacts. The objective is to provide a clear and reproducible map of how this codebase turns chemistry specifications into variational energies and error metrics.**

***Index Terms*—quantum chemistry, variational quantum eigensolver, ADAPT-VQE, PennyLane, PyTorch Lightning**

This document is process-oriented and explains three things directly:

- what inputs are provided,
- what transformations are applied,
- what outputs and quantitative results are produced.

The workflow follows the VQE formulation [1] and ADAPT-VQE ansatz growth [2], with Jordan-Wigner qubit mapping [3] and UCC-style fermionic excitations [4].

I. Inputs

A. Molecular and Chemistry Inputs

The molecule is split across two files:

- molecules/LiH.xyz (geometry): Li at (0, 0, 0) and H at (0, 0, 1.5712755877) Angstrom.
- molecules/LiH.yaml (chemistry metadata): charge = 0, multiplicity = 1, basis_name = sto-3g, mapping = jordan_wigner, method = dhf, active_electrons = 2, active_orbitals = 4.

The electronic structure target can be summarized as the ground-state problem $E_0 = \min_\psi \psi^\dagger H \psi$, where H is the second-quantized molecular Hamiltonian.

B. Runtime and Optimization Inputs

Default runtime parameters (from src/app/settings.py) are:

- seed = 2002
- vqe.algorithm = "adapt"
- vqe.common.device_name = "lightning.qubit"
- vqe.common.lbfgs_max_iter = 10
- vqe.adapt.max_steps = 1
- vqe.adapt.grad_tol = 3e-3
- vqe.adapt.finite_diff_eps = 1e-3
- vqe.adapt.pretrain_maxiter = 30
- train.max_epochs = 1
- train.steps_per_epoch = 100
- train.optimizer = "lbfgs"
- train.learning_rate = 0.05

II. Transformations

A. 1) Build the Hamiltonian and Reference State

src/app/chem/molecule_specs.py loads and validates molecule data, then src/app/chem/hamiltonians.py constructs:

- qubit Hamiltonian H_q,
- number of qubits n,
- number of active electrons N_e,
- Hartree-Fock bitstring |phi_HF> used as reference.

The fermionic Hamiltonian has the standard form $H_f = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s$, then is mapped (Jordan-Wigner [3]) to $H_q = \sum_{k=1}^{M} c_k P_k, \quad P_k \in \{I, X, Y, Z\}^n$.

The exact target energy is computed by sparse diagonalization (eigsh, SciPy [5]): $E_{\text{exact}} = \min_\psi \psi^\dagger H_q \psi$.

B. 2) Build the ADAPT Operator Pool and Select New Operators

src/app/adapt_vqe_module.py creates an excitation pool from qml.qchem.excitations: $P = \{\tau_i\}_{i=1}^{N_{\text{pool}}}$.

At each ADAPT step, each candidate is scored with a finite-difference gradient proxy $g_i \approx \frac{E(\theta, +\varepsilon e_i) - E(\theta, -\varepsilon e_i)}{2\varepsilon}$.

The selected operator is $i^* = \arg\max_i |g_i|$, and the ansatz is extended as $U^{t+1}(\theta) = \exp(\theta_{t+1} \tau_{i^*}) U^t(\theta)$.

After insertion, parameters are pre-optimized with L-BFGS-B (SciPy) before Lightning training.

C. 3) Outer Optimization in Lightning

The model objective is the absolute energy gap to the exact solver: $L(\theta) = |E(\theta) - E_{\text{exact}}|$.

Training runs over a synthetic step dataset (src/app/steps_data_module.py) so optimization length is controlled by steps_per_epoch. Checkpoints and metrics are emitted by callbacks configured in src/app/trainer.py.

III. Outputs

A. Static Outputs Produced During Setup

For the default LiH configuration, module construction produces:

- n_qubits = 8
- n_electrons = 2
- hf_state = [1, 1, 0, 0, 0, 0, 0, 0]
- Hamiltonian terms M = 105
- Hilbert-space dimension 2^8 = 256
- ADAPT pool size N_pool = 15
- selected first ADAPT operator: ("double", ((0, 1), (4, 5)))

- strongest initial gradient magnitude: |g_i| = 0.023603439331054688

B. Training and Artifact Outputs

Produced files include:

- lightning_logs/version_*/metrics.csv (per-step logged losses),
- lightning_logs/version_*/checkpoints/*.ckpt (model states),
- optional W&B run directories under wandb/ and lightning_template_vqe/.

C. Numerical Results from Repository Checkpoints

Reference exact value from the stored checkpoints:

- E_exact = -7.8644891084057305 Hartree.

Initial (Hartree-Fock / zero-parameter) value for the selected one-operator ADAPT ansatz:

- E_HF = -7.86266565322876 Hartree,
- |E_HF - E_exact| = 0.001823455176967137 Hartree.

Checkpointed ADAPT values:

- lightning_logs/version_0/checkpoints/vqe-step=00002-train_loss=0.001717.ckpt: theta = 0.004659244527978341, E = -7.862771987915039, |E - E_exact| = 0.001717120490677182 Hartree.
- lightning_logs/version_1/checkpoints/vqe-step=00003-train_loss=0.001646.ckpt: theta = 0.00796484071212286, E = -7.862843036651611, |E - E_exact| = 0.0016460717541191272 Hartree.

So, for this current one-step ADAPT setup, optimization improves the energy gap relative to the HF starting point, but does not yet reach chemical-accuracy scale. This is consistent with ADAPT-VQE behavior when ansatz depth is intentionally limited [2].

IV. Process Summary

End-to-end, the pipeline is:

1) Load geometry + chemistry metadata.
2) Build molecular Hamiltonian and HF state.
3) Map to qubits and compute exact reference energy.
4) Build ADAPT pool, score candidates, append best operator.
5) Pre-optimize with L-BFGS-B.
6) Train remaining parameters with Lightning/LBFGS on step-index data.
7) Emit energies, losses, checkpoints, and optional experiment logs.

This format is intended to make future updates straightforward: each new molecule or optimization setting can be documented by updating the same input/process/output slots.

References

[1] A. Peruzzo *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, p. 4213, 2014, doi: 10.1038/ncomms5213.

[2] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," *Nature Communications*, vol. 10, no. 1, p. 3007, 2019, doi: 10.1038/s41467-019-10988-2.

[3] P. Jordan and E. Wigner, "Über das Paulische Äquivalenzverbot," *Zeitschrift für Physik*, vol. 47, no. 9–10, pp. 631–651, 1928, doi: 10.1007/BF01331938.

[4] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, "Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz," *Quantum Science and Technology*, vol. 4, no. 1, p. 14008, 2018, doi: 10.1088/2058-9565/aad3e4.

[5] P. Virtanen *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.