Quantum Environment Is All You Need

Reinforcement Learning Agents Learn Chemistry

from Real Quantum Eigensolver Computations

Built on the KANAD Governance-Driven Quantum Chemistry Framework

Deeprealm Labs

https://deeprealm.in

Abstract. What if an RL agent could learn chemistry not from textbooks or force fields, but from the Schrödinger equation itself? We present Gymnasium environments where every reward signal is a real VQE quantum computation. In a controlled experiment on H_2, an energy-gradient reward produces monotonic learning improvement—71% → 94% near-equilibrium rate, 5.4× error reduction—while an exploration reward shows no learning under identical conditions. Built on the KANAD governance-driven quantum chemistry framework (49× faster than generic ansatze), with 93% cache hit rate at 4,096 timesteps. Early-stage, honest, and working.

Three Contributions

1. **Quantum RL environments**—four Gymnasium envs backed by real VQE, from bond stretching to reaction exploration
2. **Reward design matters**—controlled experiment proving energy-gradient rewards learn, exploration rewards don't
3. **Feasible today**—4,096 VQE-backed timesteps in under 10 minutes on a laptop, 93% cache hit rate

1 THE IDEA

Every RL environment for molecular discovery uses the same shortcut: classical energy models, force fields, or pre-computed lookup tables as the reward signal [Zhou et al., 2019, Simm et al., 2020]. The agent never touches real quantum mechanics—it learns an approximation of an approximation.

We remove the shortcut entirely. Our agent interacts with the Variational Quantum Eigensolver (VQE) [Peruzzo et al., 2014]—an algorithm that solves the electronic Schrödinger equation on a quantum circuit. Every energy the agent receives comes from:

$$E(\boldsymbol{\theta}) = \langle \psi(\boldsymbol{\theta}) | H | \psi(\boldsymbol{\theta}) \rangle \geq E_0 \tag{1}$$

Not a neural network surrogate. Not a lookup table. The actual quantum mechanical ground state energy, computed fresh for whatever molecular configuration the agent creates.

2 HOW IT WORKS

KANAD is a governance-driven quantum chemistry framework. Its key idea: *the type of bonding should dictate the quantum circuit*, not the other way around. Given two atoms, KANAD auto-detects whether the bond is covalent, ionic, or metallic from electronegativity differences, then constrains the VQE circuit to physically relevant operations via governance protocols.

This governance achieves **49× fewer function evaluations** than generic ansatze [Kandala et al., 2017] (∼20 vs. ∼1,000 for H_2), which is what makes real-quantum RL feasible: each VQE call takes ∼0.5s instead of ∼25s.

2.1 Four Environments

Each targets a different aspect of chemistry:

DissociationEnv— Stretch and compress a diatomic bond. Learn potential energy surfaces: where bonds are stable, where they break.

GeometryOptEnv— Nudge atoms in 3D to find the lowest-energy geometry. Learn equilibrium structures.

MoleculeBuilderEnv— Place atoms one at a time. Learn which elements bond, at what distances, and why noble gases don't.

ReactionExplorerEnv— Walk along a reaction coordinate.

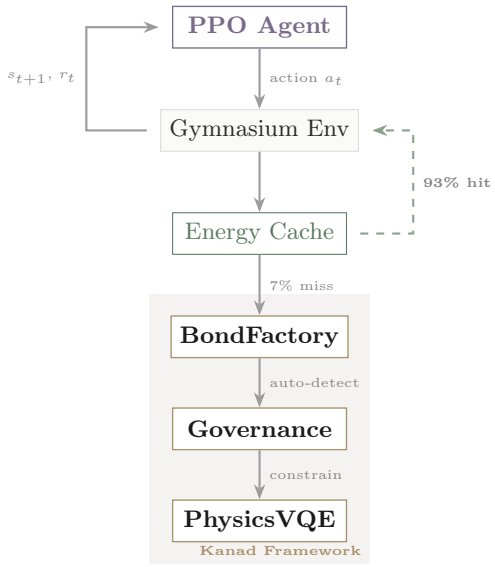

Figure 1: Every action triggers a quantum computation. The agent proposes a molecular configuration; KANAD's governance-driven solver computes the exact energy; a cache eliminates 93% of redundant VQE calls.

Table 1: Identical configuration for both agents.

Parameter	Value
Algorithm	PPO [Schulman et al., 2017], seed 42
Timesteps	4,096 (≈136 episodes)
Molecule	H_2 (4 qubits, STO-3G)
Episode length	30 steps
Solver	PhysicsVQE (statevector)

Learn where transition states and energy barriers live.

All share a 50-dimensional observation vector encoding atomic properties (electronegativity, valence electrons, positions) and quantum state (energy, convergence, correlation energy). Standard Gymnasium API—works with any RL library.

3 THE EXPERIMENT

Does the agent actually learn? We designed a controlled experiment to find out—same agent, same molecule, same hyperparameters, *only the reward function differs.*

3.1 Setup

Two reward functions compete on DissociationEnv with H_2:

Energy-gradient reward gives immediate per-step feedback—"did this action lower the energy?"

$$r_t = \tanh(10 \cdot \Delta E_t) + \mathbf{1}[E_t < E_t^*] \cdot 0.5 - 0.02 \qquad (2)$$

where ΔE_t is the energy decrease from the previous step and

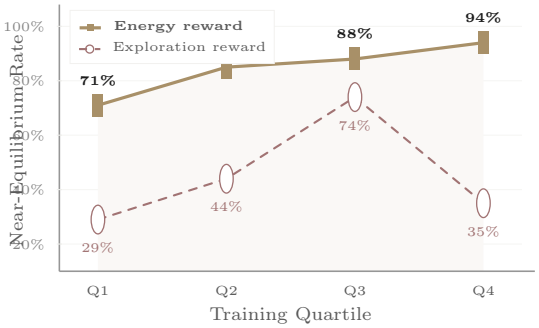

Figure 2: Controlled experiment. Same agent, molecule, hyperparameters—only the reward differs. Energy-gradient reward (solid, **71%→94%**) shows monotonic improvement; exploration reward (dashed) shows no learning trend.

E_t^* is the running best. Unconverged VQE gives $r_t = -1$.

Exploration reward evaluates trajectory quality—PES smoothness, coverage, detection of energy minima. Physically motivated, but rewards the *path*, not the *decision*.

3.2 Results

We divide each 136-episode training run into four quartiles (Q1–Q4) and measure how often the agent finds configurations within 0.15 Å of the true equilibrium ($r_{eq} = 0.74$ Å).

Central Result

Energy-gradient reward learns. Near-equilibrium rate rises monotonically: 71% → 85% → 88% → **94%**. Distance error drops **5.4**× (0.258 → 0.048 Å). The deterministic policy converges consistently to $r \approx 1.4$ Å across all evaluation rollouts.

Exploration reward does not learn. Near-equilibrium rate fluctuates: 29% → 44% → 74% → 35%. No monotonic trend. The deterministic policy drifts to $r \approx 2.8$ Å with high variance.

Table 2: Distance error by quartile (Å). Lower is better.

Reward	Q1	Q2	Q3	Q4	Trend
Energy	0.258	0.105	0.072	**0.048**	↓ 5.4×
Exploration	0.858	0.615	0.488	0.615	—

3.3 Why One Works and the Other Doesn't

The energy-gradient reward gives *per-step* feedback: "this action lowered the energy by ΔE, here's your reward proportional to that." PPO can directly optimize this signal.

The exploration reward evaluates *trajectory-level* properties: "this episode produced a smooth PES scan." But any random walk through the PES produces a reasonably smooth curve—the reward provides no gradient distinguishing good policy decisions from random ones.

Design rewards around *local energy gradients*, not global trajectory properties. The quantum simulator gives you per-step energy—use it as per-step feedback.

3.4 Infrastructure

Table 3: Training runs on a consumer laptop (no GPU needed).

	Energy	Exploration
Wall time	9.3 min	5.7 min
Unique VQE calls	306	306
Cache hit rate	**93.0%**	92.3%
Best energy found	−1.1373 Ha (<0.02 mHa from FCI)	

93% cache hit rate means only 306 of 4,096 timesteps triggered a fresh VQE computation. Governance makes each VQE call ∼0.5s (vs. ∼25s without). Together: real-quantum RL on a laptop.

4 HONEST ASSESSMENT

- **Not converged.** The learned policy reaches $r \approx 1.4$ Å, not 0.74 Å. With ∼60 gradient updates, the agent learned "lower energy is better" but hasn't had enough training to reach the exact minimum.
- **H_2 only.** We validate on 4 qubits. LiH (12 qubits, ∼14s/eval) and H_2O (14 qubits, ∼30s/eval) are accessible but require hours of training.
- **One env validated.** Four environment types exist and pass API tests, but only DissociationEnv has RL training results.

- **Genuine learning from quantum mechanics.** Monotonic improvement across quartiles with measurable policy change.
- **Reward design is the critical variable.** The controlled experiment cleanly isolates this—not the quantum computation, not the agent architecture.
- **Practical on current hardware.** Governance + caching makes the quantum simulator fast enough to be an RL reward signal.

5 RELATED WORK

RL for molecular design with classical models [Zhou et al., 2019, Simm et al., 2020]; RL for quantum control [Bukov et al., 2018, Fösel et al., 2018]; active learning over chemical space [von Lilienfeld et al., 2020, Smith et al., 2018]; adaptive VQE [Grimsley et al., 2019, Tang et al., 2021]; physics-informed ML [Karniadakis et al., 2021]. Our work differs: the RL reward is a live VQE computation, not a classical model, neural surrogate, or pre-computed table. KANAD's governance protocols make this feasible by reducing VQE cost 49×.

6 CONCLUSION

We built Gymnasium environments where the reward signal is the Schrödinger equation, solved live by VQE. A controlled experiment shows the concept works: energy-gradient rewards produce genuine learning (71% → 94% near-equilibrium, 5.4× error reduction), while exploration rewards produce none.

This is early-stage work. The agent hasn't converged to the exact equilibrium. Only H_2 is validated. One of four environments has training results. We present it honestly: a proof of concept that *real-quantum RL is feasible, and reward design is what makes it work.*

The building blocks are in place: governance-driven solvers (49× faster), energy caching (93% hit rate), modular environments (standard Gymnasium API), and a curriculum from 4 to 14 qubits. As quantum hardware scales beyond classical simulation limits, agents trained this way will discover chemistry where no textbook answers exist.

The quantum environment is all you need—
but the reward function is what makes it work.

References

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific Reports*, 9(1):10752, 2019.

Gregor NC Simm, Robert Pinsler, and José Miguel Hernández-Lobato. Reinforcement learning for molecular design guided by quantum mechanics. *Proceedings of the 37th International Conference on Machine Learning*, pages 8959–8969, 2020.

Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5: 4213, 2014.

Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small

molecules and quantum magnets. *Nature*, 549(7671):242–246, 2017.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Marin Bukov, Alexandre GR Day, Dries Sels, Phillip Weinberg, Anatoli Polkovnikov, and Pankaj Mehta. Reinforcement learning in different phases of quantum control. *Physical Review X*, 8(3):031086, 2018.

Thomas Fösel, Petru Tighineanu, Talitha Weiss, and Florian Marquardt. Reinforcement learning with neural networks for quantum feedback. *Physical Review X*, 8(3):031084, 2018.

O Anatole von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Exploring chemical compound space with quantum-based machine learning. *Nature Reviews Chemistry*, 4(7):347–358, 2020.

Justin S Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E Roitberg. Less is more: Sampling chemical space with active learning. *The Journal of Chemical Physics*, 148(24):241733, 2018.

Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature Communications*, 10(1):3007, 2019.

Ho Lun Tang, VO Shkolnikov, George S Barron, Harper R Grimsley, Nicholas J Mayhall, Edwin Barnes, and Sophia E Economou. qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor. *PRX Quantum*, 2(2):020310, 2021.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.