KANAD: A Governance-Driven Quantum Chemistry Framework for the NISQ Era

---

Technical White Paper v1.0

Authors

Mukul | Vivek

DeepRealm Labs

Contact: mk@deeprealm.in

Abstract

Kanad is a quantum chemistry framework that introduces a fundamental paradigm shift: instead of applying uniform quantum algorithms to all molecules, it recognizes that ionic, covalent, and metallic bonds exhibit fundamentally different electronic behaviors requiring distinct quantum representations. This governance-driven multi-representation architecture automatically configures quantum circuits, Hamiltonians, and fermion-to-qubit mappings based on the physics of chemical bonding. Built for the NISQ (Noisy Intermediate-Scale Quantum) era, Kanad provides researchers with physics-informed quantum simulation tools that reduce circuit complexity while maintaining chemical relevance. This white paper presents the theoretical foundations, technical architecture, and practical applications of the Kanad framework.

Table of Contentsimage.png

---

1. Introduction

1.1 The Molecular Simulation Challenge

Simulating molecular systems lies at the heart of chemistry, drug discovery, and materials science. The electronic structure problem—determining how electrons arrange themselves around nuclei—requires solving the Schrödinger equation, which becomes exponentially complex as molecules grow. For a molecule with N electrons, the full configuration space scales as O(2^N), quickly exceeding classical computational limits.

Quantum computers offer a natural solution: qubits can exist in superpositions and become entangled, mirroring the quantum behavior of electrons. However, current quantum hardware operates in the NISQ regime—limited qubits, significant noise, and shallow circuit depths. Making practical use of these devices requires algorithms that are both physically meaningful and hardware-efficient.

1.2 The Governance Innovation

Kanad's central insight is that the physics of chemical bonding should dictate the quantum algorithm.

Traditional quantum chemistry frameworks apply identical algorithms to all molecules:

- Sodium chloride (ionic) receives the same treatment as hydrogen gas (covalent)
- A metal lattice uses the same circuit topology as a water molecule

This uniformity wastes quantum resources. An ionic bond, where electrons are localized on individual ions, does not require the extensive entanglement needed for delocalized metallic electrons.

Kanad implements governance protocols—rule-based systems that analyze chemical bonding characteristics and automatically configure:

Component	What Governance Determines
Representation	How electron orbitals map to quantum states
Hamiltonian	Which energy terms are most significant
Mapping	How fermionic operators become qubit gates
Ansatz	The structure of variational quantum circuits
Entanglement	Which qubits should be correlated and how

1.3 Design Philosophy

Kanad is built on three principles:

1. Physics-First: Chemical knowledge guides algorithm design, not vice versa
2. NISQ-Aware: Circuits are designed for current hardware limitations
3. Research-Oriented: Flexibility for exploration rather than black-box execution

---

2. Theoretical Foundations

2.1 The Electronic Structure Problem

The time-independent Schrödinger equation describes molecular electronic structure:

```
Ĥ|Ψ⟩ = E|Ψ⟩
```

For a molecule with M nuclei and N electrons, the electronic Hamiltonian (in atomic units, under the Born-Oppenheimer approximation) is:

```
Ĥ_elec = -½ Σᵢ ∇ᵢ² - Σᵢₐ (Zₐ/rᵢₐ) + Σᵢ<ⱼ (1/rᵢⱼ)
```

The three terms represent:

- Kinetic energy: Electron motion
- Nuclear attraction: Coulomb attraction between electrons and nuclei
- Electron repulsion: Coulomb repulsion between electron pairs

The nuclear repulsion energy V_nn is added as a constant for fixed geometry.

2.2 Second Quantization

Kanad uses second quantization, describing many-electron systems through creation (â†) and annihilation (â) operators satisfying fermionic anticommutation:

$$\{\hat{a}_i, \hat{a}_j^\dagger\} = \delta_{ij}$$

$$\{\hat{a}_i, \hat{a}_j\} = \{\hat{a}_i^\dagger, \hat{a}_j^\dagger\} = 0$$

The electronic Hamiltonian becomes:

$$\hat{H} = \Sigma_{p\gamma}\ h_{p\gamma}\ \hat{a}_p^\dagger \hat{a}_\gamma + \tfrac{1}{2}\ \Sigma_{p\gamma rs}\ g_{p\gamma rs}\ \hat{a}_p^\dagger \hat{a}_\gamma^\dagger \hat{a}_s \hat{a}_r + V\_nn$$

Where h_pq are one-electron integrals and g_pqrs are two-electron integrals, computed using PySCF.

2.3 Basis Sets

Molecular orbitals are expanded as linear combinations of atomic orbitals (LCAO):

$$\varphi_i(r) = \Sigma\mu\ C_{\mu i}\ \chi\mu(r)$$

Kanad supports standard Gaussian basis sets:

Basis	Description	Use Case
STO-3G	Minimal basis	Quick exploration

Basis	Description	Use Case
3-21G	Split-valence	Balanced accuracy
6-31G	Enhanced split-valence	Production calculations
6-31G(d,p)	With polarization	High accuracy
cc-pVDZ	Correlation-consistent	Benchmark quality

---

3. Quantum Computing for Chemistry

3.1 Qubits and Superposition

Classical bits exist as 0 or 1. Qubits exist in superpositions:

```
|ψ⟩ = α|0⟩ + β|1⟩, where |α|² + |β|² = 1
```

For N qubits, the state space dimension is 2^N, matching the exponential scaling of electronic structure.

3.2 Fermionic-to-Qubit Mappings

Electrons are fermions; qubits are not. Kanad implements multiple mappings:

Jordan-Wigner Transformation:

```
âⱼ† → ½(Xⱼ - iYⱼ) ⊗ Z₀Z₁...Zⱼ₋₁
```

- Natural for localized systems (ionic bonds)
- Z-string enforces anticommutation

Bravyi-Kitaev Transformation:

- Tree-like parity encoding
- Shorter Pauli strings: O(log N) vs O(N)
- Better for delocalized systems (covalent bonds)

Governance-Based Selection:

Kanad automatically selects the optimal mapping based on bonding type—Jordan-Wigner for ionic (locality preserved), Bravyi-Kitaev for covalent (efficiency for delocalized orbitals).

3.3 The Variational Principle

The variational quantum eigensolver (VQE) exploits:

```
E(θ) = ⟨Ψ(θ)|Ĥ|Ψ(θ)⟩ ≥ E₀
```

Any trial wavefunction provides an upper bound to the true ground state energy. VQE iteratively optimizes parameters θ to minimize this bound.

---

4. Chemical Bonding and Classification

4.1 Electronegativity-Based Detection

Kanad classifies bonds using Pauling electronegativity differences (ΔEN):

```
# From kanad/bonds/bond_factory.py

EN_IONIC_THRESHOLD = 1.7 # ΔEN > 1.7 → ionic

EN_POLAR_THRESHOLD = 0.4 # ΔEN > 0.4 → polar covalent
```

Classification Logic:

```
IF both atoms are metals:

→ METALLIC bonding

ELSE IF ΔEN > 1.7:

→ IONIC bonding

ELSE:
```

```
→ COVALENT bonding
```

4.2 Ionic Bonding

Physical Character:

- Electron transfer from low-EN to high-EN atom
- Localized charges on ionic sites
- Electrostatic attraction dominates

Example: NaCl (ΔEN = 2.23)

- Na (EN = 0.93) → $Na^+ + e^-$
- Cl (EN = 3.16) + e^- → Cl^-

Quantum Requirements:

- Localized orbital representation
- Minimal inter-site entanglement
- Strong on-site correlations (Hubbard U)

4.3 Covalent Bonding

Physical Character:

- Electron sharing through orbital overlap
- Bonding/antibonding molecular orbital formation
- Hybridization (sp, sp^2, sp^3)

Example: H_2 (ΔEN = 0.0)

- σ bonding orbital: $\psi = (\varphi_A + \varphi_B)/\sqrt{2}$
- σ *antibonding:* $\psi = (\varphi_A - \varphi_B)/\sqrt{2}$

Quantum Requirements:

- Molecular orbital basis
- Paired entanglement (Bell states for electron pairs)
- Explicit correlation for bond breaking

4.4 Metallic Bonding

Physical Character:

- Delocalized "electron sea"
- Band structure with Fermi surface
- Collective electronic behavior

Example: Fe-Fe in BCC lattice

Quantum Requirements:

- k-space (momentum) representation
- Global/collective entanglement
- Periodic boundary conditions

---

5. The Governance Protocol System

5.1 Architecture

Each governance protocol consists of prioritized rules:

```
# From kanad/governance/protocols/base_protocol.py

@dataclass

class GovernanceRule:

name: str # Rule identifier

description: str # Human-readable explanation

condition: Callable # When rule applies

action: Callable # What modification to make

priority: int # Execution order (lower = earlier)

required: bool # Must be satisfied
```

Rules are applied sequentially during circuit construction, with higher-priority rules executed first.

5.2 Ionic Governance Protocol

Design Principle: Minimize unnecessary entanglement; preserve charge localization.

Implemented Rules:

Rule	Priority	Description
`localized_gates_only`	100	Enforce nearest-neighbor interactions only
`sparse_connectivity`	90	Average qubit degree < 2
`particle_number_conservation`	80	Preserve electron count per site
`forbid_collective_gates`	70	No gates spanning > 2 qubits

Entanglement Strategy: Minimal (nearest-neighbor only)

5.3 Covalent Governance Protocol

Design Principle: Capture bonding/antibonding correlation through paired entanglement.

Implemented Rules:

Rule	Priority	Description
`hybridization_first`	100	Check orbital hybridization before bonding
`molecular_orbital_formation`	90	Create bonding/antibonding MO pairs
`electron_pair_entanglement`	80	Bell states for electron pairs
`spin_symmetry`	70	Maintain singlet states
`no_long_range_entanglement`	60	Restrict to bonding pairs

Entanglement Strategy: Paired (Bell-pair structure)

5.4 Metallic Governance Protocol

Design Principle: Preserve collective delocalization and band structure.

Implemented Rules:

Rule	Parameter	Description
`delocalization`	min_score: 0.3	Ensure electron delocalization

Rule	Parameter	Description
`fermi_surface`	DOS(E_F) > 0	Validate metallic character
`periodic_bc`	min_atoms: 3	Periodic boundary conditions
`mott_transition`	max U/t: 4.0	Avoid insulating transition

Entanglement Strategy: Collective (GHZ-like states)

5.5 How Governance Reduces Circuit Complexity

By encoding physical constraints directly into circuit construction, governance eliminates operations that would be physically meaningless:

- Ionic: No long-range entanglement gates (electrons don't delocalize)
- Covalent: Only bonding orbital pairs entangle (chemical intuition)
- Metallic: Global operations reflect collective behavior

This physics-informed reduction yields shallower circuits without sacrificing chemical accuracy.

---

6. Bond-Specific Hamiltonians

6.1 Covalent Hamiltonian

Implementation: `kanad/core/hamiltonians/covalent_hamiltonian.py`

Standard molecular electronic Hamiltonian with hybridization support:

```
Ĥ_covalent = Σ_μν h_μν ĉ†_μ ĉ_ν + ½ Σ_μνλσ (μν|λσ) ĉ†_μ ĉ†_ν ĉ_σ ĉ_λ
```

Features:

- PySCF integration for integral computation
- Active space support for larger molecules
- Hybridization-aware orbital ordering

6.2 Ionic Hamiltonian

Implementation: `kanad/core/hamiltonians/ionic_hamiltonian.py`

Hubbard-like model emphasizing localized interactions:

```
Ĥ_ionic = Σᵢ εᵢ n̂ᵢ + Σ_⟨ij⟩ tᵢⱼ (â†ᵢ âⱼ + h.c.) + Σᵢ Uᵢ n̂ᵢ↑ n̂ᵢ↓
```

Parameters:

- ε_i: On-site energy (from electronegativity)
- t_{ij}: Transfer integrals (weak for ionic systems)
- U_i: Hubbard U (on-site repulsion)

6.3 Metallic Hamiltonian

Implementation: `kanad/core/hamiltonians/metallic_hamiltonian.py`

Tight-binding model with periodic boundary conditions:

```
Ĥ_metal = Σ_⟨ij⟩,σ tᵢⱼ (â†ᵢσ âⱼσ + h.c.) + Σᵢ,σ εᵢ n̂ᵢσ + U Σᵢ n̂ᵢ↑ n̂ᵢ↓
```

Features:

- Hopping parameter t controls delocalization
- Periodic boundary conditions for lattice symmetry
- Fermi surface physics

6.4 Key Differences Summary

Aspect	Covalent	Ionic	Metallic
Basis	Molecular orbitals	Localized atomic	k-space (momentum)
Dominant Terms	Overlap integrals	On-site energy, Hubbard U	Hopping t
Electron Behavior	Shared (bonding MOs)	Transferred (localized)	Delocalized (bands)
Correlation	Bonding pairs	Site-specific	Collective

---

7. Quantum Solvers

7.1 Variational Quantum Eigensolver (VQE)

Implementation: `kanad/solvers/vqe_solver.py`

Algorithm:

1. Prepare parameterized trial state $|\Psi(\theta)\rangle$
2. Measure expectation value $\langle \hat{H} \rangle$
3. Classically optimize θ to minimize energy
4. Repeat until convergence

Supported Optimizers:

- `SLSQP` (default): Sequential Least Squares, gradient-based
- `COBYLA`: Gradient-free, robust to noise
- `L-BFGS-B`: Limited-memory BFGS with bounds

Supported Ansätze:

- UCC (Unitary Coupled Cluster): Chemistry gold standard, includes singles and doubles excitations
- Hardware-Efficient: Native to device connectivity, 3 layers with linear entanglement
- Governance-Aware: Physics-informed circuits adapted to bonding type

Backends:

- `statevector`: Exact classical simulation
- `qasm` / `aer`: Shot-based Qiskit simulation
- `ibm`: IBM Quantum hardware via Qiskit Runtime
- `bluequbit`: GPU-accelerated cloud simulation

Convergence: Energy threshold (default: 10^{-6} Hartree) with optimizer-specific criteria.

7.2 Subspace Quantum Diagonalization (SQD)

Implementation: `kanad/solvers/sqd_solver.py`

Alternative to VQE that constructs and diagonalizes a subspace Hamiltonian:

1. Generate basis states (bitstrings, random states, or from VQE)

2. Measure Hamiltonian matrix elements $\langle\psi_i|\hat{H}|\psi_j\rangle$
3. Classically diagonalize the subspace matrix
4. Extract eigenvalues and eigenvectors

Default Shots: 8192 per measurement circuit

Advantages:

- Can find multiple eigenvalues simultaneously
- More robust to barren plateaus
- Natural for excited states

7.3 Excited States Solver

Implementation: `kanad/solvers/excited_states_solver.py`

Implemented Methods:

Method	Status	Description
CIS	Complete	Configuration Interaction Singles
TDDFT	Complete	Time-Dependent DFT approximation
VQE	Complete	Orthogonally-constrained variational
SQD	Complete	Subspace diagonalization for excited states
QPE	Planned	Quantum Phase Estimation (future)

7.4 Hi-VQE Mode

For larger systems, Kanad supports hierarchical VQE:

- Active space reduction
- Iterative subspace expansion
- Significantly reduced measurement counts

---

8. Molecular Dynamics

8.1 Classical Molecular Dynamics

Implementation: `kanad/dynamics/md_simulator.py`

Molecular dynamics integrates Newton's equations:

$$m_i (d^2R_i/dt^2) = F_i = -\nabla_i V(R)$$

Integrators:

- Velocity Verlet (default): Symplectic, time-reversible, $O(\Delta t^2)$
- Leapfrog: Equivalent to Verlet with staggered variables
- Runge-Kutta 4: Higher accuracy, non-symplectic

Thermostats:

- Berendsen: Simple velocity rescaling
- Nosé-Hoover: Correct canonical ensemble
- Langevin: Stochastic with friction term

8.2 Quantum-Enhanced Molecular Dynamics

Implementation: `kanad/dynamics/quantum_md.py`

Force Methods:

- `hf`: Hartree-Fock forces
- `mp2`: Second-order perturbation theory
- `vqe`: VQE-computed forces
- `hivqe`: Hierarchical VQE forces
- `sqd`: SQD-derived forces

Current Implementation:

Quantum forces are computed via numerical gradients with solver caching for efficiency. The framework supports analytical gradients through the parameter shift rule (structure implemented, full deployment ongoing).

Note: Quantum force evaluation is computationally expensive. Solver caching provides significant speedup for repeated calculations along trajectories.

---

9. Environmental Effects

9.1 Solvent Models

Implementation: `kanad/environment/solvent.py`

Supported Models:

Polarizable Continuum Model (PCM):

```
ΔG_elec = -½ q²(1 - 1/ε)/a
```

- Born model with reaction field
- Dielectric screening of Coulomb interactions

SMD (Solvation Model based on Density):

- PCM electrostatics
- Abraham descriptor corrections for hydrogen bonding
- Surface tension contributions

Solvent Database (12 solvents):

Solvent	ε (dielectric)	Applications
Water	78.4	Biological systems
DMSO	46.7	Drug solubility
Acetonitrile	37.5	Electrochemistry
Methanol	32.7	Organic synthesis
Chloroform	4.81	Extraction
Hexane	1.88	Nonpolar reference

9.2 Additional Environmental Terms

- Cavity formation energy: Scaled Particle Theory
- Dispersion interactions: Based on refractive index
- Non-electrostatic contributions: Empirically calibrated

---

10. Molecular Properties and Spectroscopy

10.1 Ground State Properties

Kanad computes:

- Dipole moment: From electron density
- Mulliken charges: Population analysis
- Bond order: From density matrix
- HOMO-LUMO gap: Frontier orbital energies

10.2 Spectroscopic Properties

UV-Vis Absorption:

- Excitation energies from excited state solver
- Oscillator strengths for transition intensities
- Spectrum generation with Gaussian broadening

Vibrational Analysis:

- Normal mode frequencies from Hessian
- IR intensities (dipole derivatives)
- Thermochemical corrections

10.3 Thermochemistry

- Zero-point energy (ZPE): Quantum vibrational ground state
- Thermal corrections: Translational, rotational, vibrational
- Gibbs free energy: G = H - TS at specified temperature

10.4 Active Space Selection

Implementation: `kanad/solvers/active_space.py`

Strategies:

- `homo_lumo`: Window around frontier orbitals
- `natural_orbitals`: Based on occupation numbers
- `governance`: Protocol-guided selection

Active space selection enables treatment of larger molecules by focusing quantum resources on chemically relevant orbitals.

---

11. Technical Architecture

11.1 Module Structure

```
kanad/

├── bonds/ # Bond factory and implementations

│ ├── bond_factory.py # User-facing API

│ ├── covalent_bond.py

│ ├── ionic_bond.py

│ └── metallic_bond.py

├── governance/ # Protocol system

│ └── protocols/ # Ionic, covalent, metallic protocols

├── core/

│ ├── hamiltonians/ # Bond-specific Hamiltonians

│ ├── mappers/ # JW, BK transformations

│ ├── representations/# Second quantization, LCAO

│ ├── integrals/ # One/two-electron integrals

│ └── constants/ # Physical constants, atomic data

├── solvers/ # VQE, SQD, excited states

├── ansatze/ # Circuit architectures

├── dynamics/ # Molecular dynamics

├── environment/ # Solvent models

├── analysis/ # Property calculators

└── backends/ # IBM, BlueQubit integration
```

11.2 Dependencies

Core Requirements:

- Python 3.12+
- NumPy ≥ 1.26.0
- SciPy ≥ 1.12.0
- Qiskit ≥ 1.2.0
- PySCF ≥ 2.6.0
- OpenFermion ≥ 1.6.0

Optional:

- Qiskit IBM Runtime (cloud quantum)
- BlueQubit SDK (GPU acceleration)
- Matplotlib/Plotly (visualization)

11.3 Usage Example

```
from kanad.bonds import BondFactory
from kanad.solvers import VQESolver

# Create bond (auto-detects type from electronegativity)
bond = BondFactory.create_bond('H', 'H', distance=0.74)
print(f"Bond type: {bond.bond_type}") # 'covalent'

# Run VQE with governance-aware ansatz
solver = VQESolver(
bond=bond,
ansatz_type='governance',
optimizer='SLSQP',
```

```
max_iterations=100

)

result = solver.solve()



print(f"Ground state energy: {result['energy']:.6f} Ha")
```

---

12. Applications

12.1 Drug Discovery

Who Benefits: Pharmaceutical researchers, medicinal chemists, biotech companies

Capabilities:

- Binding energy estimation for drug-target interactions
- ADME property prediction (absorption, distribution, metabolism, excretion)
- Conformational analysis in solution (with solvent models)

12.2 Materials Science

Who Benefits: Materials scientists, battery researchers, catalyst developers

Capabilities:

- Electronic structure of novel materials
- Band structure analysis (metallic systems)
- Catalyst active site characterization

12.3 Academic Research

Who Benefits: Quantum chemistry groups, quantum computing researchers, graduate students

Capabilities:

- Testbed for new quantum algorithms
- Exploration of governance-driven approaches
- Educational demonstrations of quantum chemistry concepts

12.4 Environmental Chemistry

Who Benefits: Environmental scientists, atmospheric chemists

Capabilities:

- Reaction pathway analysis
- Solvation thermodynamics
- Pollutant degradation mechanisms

---

13. Current Limitations and Roadmap

13.1 Current State

Kanad is a research framework optimized for exploration and algorithm development. As with any NISQ-era tool, users should be aware of:

Computational Considerations:

- Quantum force calculations (VQE-based MD) are computationally intensive
- Large molecules require active space selection
- Hardware execution subject to device availability and queue times

Scope:

- Primarily validated on small molecules (H_2, LiH, H_2O, small organics)
- Mixed bonding types (e.g., organometallics) use dominant bonding character
- Relativistic effects not currently included

13.2 Roadmap

Near-term:

- Expanded benchmark validation across standard test sets
- Analytical gradient implementation for VQE forces
- Enhanced error mitigation for hardware execution

Medium-term:

- Periodic systems (solid-state materials)
- Relativistic corrections for heavy elements
- Automated active space selection

Long-term:

- Fault-tolerant algorithm readiness
- Integration with experimental automation
- Extended molecular dynamics capabilities

---

14. Conclusion

Kanad represents a physics-informed approach to quantum chemistry in the NISQ era. By recognizing that different chemical bonds require different quantum treatments, the governance-driven architecture provides:

1. Physical Relevance: Circuits encode chemical knowledge, not just mathematical optimization
2. Resource Efficiency: Physics-informed constraints reduce circuit complexity
3. Flexibility: Modular design supports research exploration and algorithm development

The framework bridges quantum computing and chemistry by embedding chemical intuition directly into quantum algorithms. For researchers exploring the intersection of these fields, Kanad provides a foundation for developing and testing new approaches to molecular simulation.

Technical Appendix

A. Physical Constants

Constant	Value	Unit
Hartree energy	27.21138	eV
Bohr radius	0.529177	Å
Speed of light	299,792,458	m/s
Planck constant	6.626×10^{-34}	J·s

B. Electronegativity Values (Pauling Scale)

Element	EN	Element	EN	Element	EN
H	2.20	C	2.55	N	3.04
O	3.44	F	3.98	Na	0.93
Cl	3.16	Li	0.98	Fe	1.83

C. Supported Metals (for Metallic Bonding Detection)

Li, Na, K, Mg, Ca, Al, Fe, Cu, Zn, Ag, Au

References

1. Peruzzo, A. et al. "A variational eigenvalue solver on a photonic quantum processor." *Nat. Commun.* 5, 4213 (2014).
2. McArdle, S. et al. "Quantum computational chemistry." *Rev. Mod. Phys.* 92, 015003 (2020).
3. Cao, Y. et al. "Quantum Chemistry in the Age of Quantum Computing." *Chem. Rev.* 119, 10856-10915 (2019).
4. Jordan, P. & Wigner, E. "Über das Paulische Äquivalenzverbot." *Z. Phys.* 47, 631-651 (1928).
5. Bravyi, S.B. & Kitaev, A.Y. "Fermionic Quantum Computation." *Ann. Phys.* 298, 210-226 (2002).

---

15. Kanad Web Interface — Coming Spring 2026

For Researchers Who Do Chemistry, Not Quantum Circuits

If you're a chemist studying reaction mechanisms, a biologist modeling protein interactions, a materials scientist searching for the next superconductor, or a physicist exploring molecular properties—you shouldn't need to learn quantum computing to benefit from it.

Kanad Web is designed for you.

What It Offers

No Installation. No Configuration. Just Science.

Visit kanad.xyz, create an account, and start running quantum chemistry calculations. The interface handles everything underneath—qubit mappings, circuit compilation, governance protocol selection, backend optimization—so you can focus on the chemistry that matters to your research.

Domain-Specific Workflows

Pre-built experimental workflows for common research tasks:

Domain	Example Workflows
Chemistry	Reaction pathway analysis, transition state search, thermochemistry
Drug Discovery	Binding affinity, ADME prediction, conformational sampling
Materials	Band structure, catalyst screening, defect formation
Biochemistry	Active site modeling, spectroscopic properties

Each workflow guides you through the scientific parameters you care about—bond lengths, solvents, temperature—not quantum parameters you shouldn't have to think about.

Run Anywhere, Pay Only for What You Use

No expensive hardware purchases. No cluster management. No GPU configurations. Your calculations run on cloud quantum simulators and real quantum hardware. You pay based on computational usage, making quantum chemistry accessible to research groups of any size.

The Vision

Quantum computing promises to transform molecular simulation, but that promise remains locked behind layers of technical complexity. Kanad Web removes those barriers.

A graduate student in a chemistry lab should be able to compute the potential energy surface of their molecule as easily as they run an NMR experiment. A pharmaceutical researcher should explore drug candidates with quantum accuracy without writing a single line of quantum code. A materials scientist should screen catalysts without debugging qubit connectivity errors.

Spring 2026 — We're building this for you.

Website: kanad.xyz

Contact: mk@deeprealm.in

Deeprealm Labs — Advancing Quantum Chemistry for the NISQ Era.