Chapter 2

QVM for Pharmacology and Drug Discovery

A Deterministic Operator-Based Computational Platform for Molecular Exploration

Karel Cápek

1 Scope and Computational Motivation

1.1 Purpose of This Whitepaper

This whitepaper specifies a domain-specialized configuration of the Quantum Virtual Machine (QVM) for use in pharmacology and drug discovery. Its purpose is to formalize a deterministic, operator-based computational framework for molecular analysis, binding evaluation, and lead optimization.

The document is written in the QFC style and is intended to serve as:

- a technical foundation for implementation,
- a reference for architectural and mathematical consistency,
- a basis for industrial-scale investment and deployment.

This is not a proposal for speculative quantum hardware, nor an extension of stochastic molecular simulation. The framework described here operates entirely within classical deterministic computation, using operator-theoretic abstractions inspired by quantum mechanics but not dependent on physical quantum effects.

1.2 Computational Nature of the Drug Discovery Problem

Drug discovery is fundamentally constrained by the structure of molecular configuration spaces. Molecules exhibit:

- high-dimensional state spaces,
- non-linear interaction structure,
- strong coupling between local and global degrees of freedom,
- sensitivity to small structural perturbations.

These properties make brute-force enumeration infeasible and render purely algorithmic approaches brittle.

Traditional computational methods attempt to address this through numerical integration of molecular dynamics, stochastic sampling techniques, or statistical learning on large empirical datasets. Each of these approaches introduces trade-offs between cost, interpretability, reproducibility, and scientific transparency.

1.3 Limitations of Existing Computational Paradigms

1.4 Classical Molecular Dynamics

Molecular dynamics approximates molecular evolution through numerical integration of force fields. While physically motivated, this approach suffers from:

- extremely high computational cost,
- slow convergence for rare or metastable configurations,
- sensitivity to integration parameters,
- limited global insight into configuration space structure.

Molecular dynamics produces trajectories rather than structural explanations.

1.5 Stochastic and Monte Carlo Methods

Stochastic sampling techniques reduce computational cost but introduce:

- irreducible randomness,
- dependence on sampling heuristics,
- difficulty in reproducing exact results,
- limited auditability.

Such properties are problematic in regulated scientific environments.

1.6 Machine Learning–Based Predictors

Machine learning methods offer speed and scalability but rely on:

- historical training data,
- opaque internal representations,
- indirect inference rather than physical modeling.

While useful for screening, these methods do not constitute a scientific model of molecular behavior and provide limited explanatory power.

1.7 Operator-Centric Reframing of Molecular Computation

This whitepaper adopts a different perspective: molecules are treated as operator systems rather than simulated objects.

In this view:

- molecular structure corresponds to a state in a structured space,

- interactions correspond to operators acting on that space,
- molecular behavior corresponds to spectral properties of composed operators.

Computation is reframed as controlled operator evolution, rather than numerical time stepping or probabilistic sampling.

This perspective aligns naturally with the QVM architecture, which is designed to execute bounded, deterministic operator programs under strict governance and audit constraints.

1.8 Determinism, Auditability, and Regulatory Alignment

A core requirement of pharmaceutical computation is long-term reproducibility and explainability. Any computational result used to guide experimental or clinical decisions must be:

- reproducible across time and hardware,
- traceable to explicit assumptions and operations,
- defensible under regulatory scrutiny.

The QVM framework enforces deterministic execution semantics, explicit operator definitions, bounded computation, and comprehensive audit traces. These properties make it suitable for integration into regulated drug development pipelines.

1.9 Scope of This Document

This whitepaper focuses on:

- mathematical abstraction of molecules as operator systems,
- definition of molecular operators relevant to pharmacology,
- execution of these operators on a QVM/QPU stack,
- interpretation of resulting spectral structures.

It does not address wet-lab validation, biological efficacy beyond molecular interaction, or clinical trial design. Those aspects are downstream consumers of the computational results produced by the system described here.

1.10 Structure of the Whitepaper

The remainder of this document is organized as follows:

- Chapter 2 formalizes the molecular state space.
- Chapter 3 defines fundamental molecular operators.
- Chapter 4 constructs the molecular Hamiltonian and its properties.
- Chapter 5 describes operator evolution and spectral analysis.
- Chapter 6 applies the framework to ligand–protein binding.
- Chapter 7 maps the model to QVM and QPU execution.
- Chapter 8 discusses computational guarantees and limitations.

Each chapter is written to be internally consistent and implementation-oriented.

1.11 Positioning Statement

The system described here is not a faster simulator, nor an approximation layer over existing tools. It is a new computational instrument for molecular science, designed around operator semantics, determinism, and explainability.

It is intended to complement experimental work, reduce exploratory cost, and provide a transparent computational foundation for rational drug design.

2 Molecular State Space and Representation

2.1 Overview

This chapter defines the mathematical representation of molecular systems within the QVM framework. The goal is to establish a deterministic, finite, and operator-compatible state space suitable for bounded execution, spectral analysis, and auditability.

Unlike classical molecular simulations, the representation adopted here does not rely on continuous trajectories or numerical integration in physical time. Instead, molecular configurations are encoded as discrete states within a structured space, upon which formally defined operators act.

The resulting state space is designed to:

- admit deterministic evolution,
- support explicit operator composition,
- remain finite and bounded by construction,
- align with QVM execution and verification constraints.

2.2 Molecule as a Structured State

A molecule is represented as a single global state composed of multiple interacting degrees of freedom. Formally, the molecular state space is defined as a structured tensor product:

$$\mathcal{H}_{\text{mol}} = \bigotimes_{k=1}^{M} \mathcal{H}_k,$$

where each subspace $\mathcal{H}_k$ corresponds to a localized molecular degree of freedom.

Typical subspaces include:

- bond-length modes,
- angular modes,
- torsional modes,
- electronic interaction proxies,
- environmental coupling parameters.

The tensor structure reflects physical locality while enabling controlled global interaction through operator coupling.

2.3 Discrete and Bounded Representation

All subspaces $\mathcal{H}_k$ are discretized explicitly. Discretization is not treated as a numerical approximation, but as a modeling decision enforced at the representational level.

Each degree of freedom admits:

- a finite basis,
- explicitly declared bounds,
- well-defined admissible ranges.

As a result, the total state space is finite-dimensional:

$$\dim(\mathcal{H}_{\text{mol}}) < \infty,$$

ensuring that all computations performed by QVM are statically bounded and verifiable.

2.4 Basis States and Physical Interpretation

Basis states of $\mathcal{H}_{\text{mol}}$ correspond to admissible molecular configurations. Each basis vector encodes:

- a specific combination of bond distances,
- angular and torsional assignments,
- electronic interaction modes,
- environmental context.

States that violate fundamental physical constraints (such as impossible steric overlap or forbidden bond geometry) are excluded from the basis by construction.

This guarantees that every representable state corresponds to a physically admissible molecular configuration.

2.5 State Normalization and Interpretation

The molecular state vector $\psi \in \mathcal{H}_{\text{mol}}$ is normalized according to:

$$\langle \psi, \psi \rangle = 1.$$

Normalization does not imply probabilistic interpretation in the sense of measurement. Instead, it serves as:

- a consistency condition for spectral analysis,
- a numerical stability requirement,
- a basis for comparing operator-induced transformations.

Observable quantities are derived from operator spectra and expectation values, not from sampling-based measurement.

2.6 State Constraints and Admissibility

Not all mathematically representable states are admissible in a given computational context. Admissibility is enforced through:

- explicit exclusion of forbidden basis states,
- projection operators eliminating invalid configurations,
- bounded parameter declarations at execution time.

Admissibility constraints are static and deterministic. No runtime correction or penalty-based enforcement is permitted.

2.7 Environmental and Contextual Encoding

Environmental factors such as solvent effects, temperature regimes, or ionic conditions are encoded as additional components of the state representation rather than as external stochastic influences.

These contextual components:

- belong to designated subspaces of $\mathcal{H}_{\text{mol}}$,
- interact with structural degrees of freedom via operators,
- remain bounded and explicitly declared.

This allows controlled exploration of environmental variation without introducing non-determinism.

2.8 Composite Molecular Systems

For interacting molecular systems, such as ligand–protein complexes, the joint state space is constructed as:

$$\mathcal{H}_{\text{total}} = \mathcal{H}_{\text{protein}} \otimes \mathcal{H}_{\text{ligand}}.$$

Interaction between subsystems is introduced exclusively through interaction operators defined in subsequent chapters. No implicit coupling is permitted at the level of state representation.

2.9 Relation to QVM Execution Semantics

The molecular state space defined in this chapter is designed to map directly onto QVM execution semantics:

- states are immutable during operator application,
- state transitions occur only at declared execution boundaries,
- all state transformations are auditable and reproducible.

This alignment ensures that molecular computation integrates seamlessly into the broader QVM/QPU stack without violating determinism or governance constraints.

2.10 Summary

This chapter has defined a finite, structured, and deterministic molecular state space suitable for operator-based computation. By enforcing admissibility, boundedness, and explicit representation, the framework establishes a stable foundation upon which molecular operators and dynamics can be rigorously defined.

The next chapter introduces the fundamental molecular operators acting on this state space and formalizes their mathematical properties.

3 Fundamental Molecular Operators

3.1 Overview

This section defines the fundamental operator classes used to model molecular structure and interaction within the QVM framework. Operators are the primary computational primitives. They act on the molecular state space defined in the previous section and encode all admissible molecular behavior.

All operators introduced here satisfy the following requirements:

- they are explicitly defined and bounded,
- they admit deterministic execution,
- they are composable under QFM rules,
- they are suitable for spectral analysis,
- they are auditable and reproducible under QVM execution.

No operator relies on continuous-time dynamics or stochastic processes.

3.2 Operator-Theoretic Perspective

Let $\mathcal{H}_{\text{mol}}$ denote the molecular state space. A molecular operator is a linear operator

$$\hat{O} : \mathcal{H}_{\text{mol}} \to \mathcal{H}_{\text{mol}}$$

with explicitly declared domain and bounds.

Operators represent physical interactions not as forces or updates, but as transformations of admissible molecular configurations. Molecular computation is therefore defined as structured operator composition.

3.3 Bond Interaction Operators

Bond interactions are represented by operators acting on bond-length subspaces. For a bonded atom pair (i, j), the bond operator is defined as:

$$\hat{B}_{ij}\psi(d_{ij}) = V_{ij}(d_{ij})\,\psi(d_{ij}),$$

where d_{ij} denotes the discretized bond-length coordinate and V_{ij} is a bounded potential function.

Bond operators:

- act locally on designated subspaces,
- encode preferred bond lengths and tolerances,
- assign zero amplitude to forbidden distances.

No numerical integration of forces is performed.

3.4 Angular Operators

Angular constraints are encoded by operators acting on angular subspaces. For a bonded triplet (i, j, k), the angular operator is given by:

$$\hat{A}_{ijk}\psi(\theta_{ijk}) = W_{ijk}(\theta_{ijk})\,\psi(\theta_{ijk}),$$

where θ_{ijk} is the discretized bond angle.

Angular operators enforce geometric constraints spectrally by suppressing inadmissible angular configurations.

3.5 Torsional Operators

Torsional degrees of freedom are modeled by operators acting on dihedral-angle subspaces. For a quadruple (i, j, k, l), the torsional operator is defined as:

$$\hat{T}_{ijkl}\psi(\phi_{ijkl}) = U_{ijkl}(\phi_{ijkl})\,\psi(\phi_{ijkl}),$$

where ϕ_{ijkl} denotes the discretized dihedral angle.

Torsional operators capture conformational flexibility while maintaining explicit bounds on admissible rotations.

3.6 Electrostatic Interaction Operators

Electrostatic interactions are represented as pairwise coupling operators between charged components:

$$\hat{E}_{ij} = q_i q_j\, K(d_{ij}),$$

where q_i and q_j are effective charges and K is a bounded interaction kernel.

Long-range interactions are encoded through operator coupling rather than explicit summation, ensuring bounded computational complexity.

3.7 Steric Exclusion Operators

Steric constraints are enforced through projection operators that eliminate physically impossible configurations. The steric exclusion operator is defined as:

$$\hat{P}_{\text{steric}} = \prod_{i<j} \left(I - \Pi_{ij}^{\text{overlap}} \right),$$

where $\Pi_{ij}^{\text{overlap}}$ projects onto states corresponding to forbidden atomic overlap.

States violating steric constraints are removed from the admissible subspace deterministically.

3.8 Environmental Interaction Operators

Environmental effects such as solvent interaction or dielectric screening are represented by external operator fields:

$$\hat{H}_{\text{env}} = \sum_k \lambda_k \hat{O}_k,$$

where $\hat{O}_k$ are environment-specific interaction operators and λ_k are bounded coupling parameters.

Environmental operators modify molecular behavior without introducing stochastic noise or hidden state.

3.9 Operator Composition

The full molecular interaction operator is constructed as a finite sum of fundamental operators:

$$\hat{H}_{\text{mol}} = \sum \hat{B}_{ij} + \sum \hat{A}_{ijk} + \sum \hat{T}_{ijkl} + \sum \hat{E}_{ij} + \hat{P}_{\text{steric}} + \hat{H}_{\text{env}}.$$

All terms are explicitly declared, bounded, and subject to validation prior to execution.

3.10 Determinism and Validation

Each operator:

- has deterministic semantics,
- admits static validation of bounds,
- cannot modify execution context or state representation.

Undeclared operators or implicit interactions are not permitted.

3.11 Summary

This section has introduced the fundamental molecular operators used in the QVM pharmacology framework. These operators form the atomic computational elements from which molecular behavior is derived.

The next section constructs the molecular Hamiltonian as a structured composition of these operators and analyzes its mathematical properties.

4 Molecular Hamiltonian Construction and Properties

4.1 Overview

This section constructs the molecular Hamiltonian as the central operator governing admissible molecular behavior within the QVM framework. The Hamiltonian is not interpreted as a physical energy operator in continuous time, but as a structured generator of constrained operator evolution suitable for deterministic execution and spectral analysis.

The Hamiltonian aggregates the fundamental molecular operators defined in the previous section into a single, analyzable object whose spectral properties encode molecular stability, interaction strength, and conformational structure.

4.2 Definition of the Molecular Hamiltonian

Let $\mathcal{H}_{\text{mol}}$ denote the molecular state space. The molecular Hamiltonian is defined as a finite, explicitly declared operator:

$$\hat{H}_{\text{mol}} = \sum_{(i,j)} \hat{B}_{ij} + \sum_{(i,j,k)} \hat{A}_{ijk} + \sum_{(i,j,k,l)} \hat{T}_{ijkl} + \sum_{(i,j)} \hat{E}_{ij} + \hat{P}_{\text{steric}} + \hat{H}_{\text{env}}.$$

Each term corresponds to a distinct interaction class and is independently bounded and validated prior to composition.

4.3 Self-Adjointness and Symmetry

All constituent operators are required to be self-adjoint or explicitly symmetrized. As a result, the molecular Hamiltonian satisfies:

$$\hat{H}_{\text{mol}} = \hat{H}_{\text{mol}}^{\dagger}.$$

Self-adjointness ensures:

- real-valued spectra,

- stable spectral decomposition,
- deterministic operator evolution.

Any operator that cannot be expressed in a self-adjoint form is inadmissible within the QVM pharmacology framework.

4.4 Boundedness and Domain Constraints

The molecular Hamiltonian is bounded on its declared domain:

$$\|\hat{H}_{\text{mol}}\| < \infty.$$

Boundedness is enforced by:

- finite discretization of all subspaces,
- explicit truncation of interaction ranges,
- projection-based elimination of forbidden states.

The domain of $\hat{H}_{\text{mol}}$ is fixed at execution time and may not be extended dynamically.

4.5 Spectral Interpretation

The spectrum of the molecular Hamiltonian,

$$\text{Spec}(\hat{H}_{\text{mol}}) = \{\lambda_n\},$$

encodes global properties of the molecular system.
In this framework:

- low spectral values correspond to stable molecular configurations,
- spectral gaps indicate robustness under perturbation,
- clustered eigenvalues indicate conformational families,
- isolated eigenvalues correspond to structurally distinct states.

No probabilistic interpretation is imposed on the spectrum.

4.6 Constraint Enforcement via Spectral Structure

Physical and chemical constraints are enforced through the spectral structure of the Hamiltonian rather than through procedural checks.
Inadmissible configurations:

- do not appear as eigenstates,
- are eliminated by projection operators,
- cannot be reached through operator evolution.

This guarantees that all spectral results correspond to physically admissible molecular states.

4.7 Hamiltonian Decomposition and Modularity

The Hamiltonian admits a modular decomposition aligned with molecular structure:

$$\hat{H}_{\text{mol}} = \hat{H}_{\text{local}} + \hat{H}_{\text{interaction}} + \hat{H}_{\text{context}}.$$

This decomposition enables:

- localized spectral analysis,
- incremental refinement of interaction models,
- controlled extension of operator libraries.

All decomposed forms are equivalent representations of the same global operator.

4.8 Stability Under Perturbation

Small perturbations of operator parameters induce bounded spectral variation:

$$\|\delta \hat{H}_{\text{mol}}\| \ll 1 \;\Rightarrow\; |\delta \lambda_n| \ll 1.$$

This property enables sensitivity analysis and rational lead optimization by evaluating how structural modifications affect spectral features.

4.9 Compatibility with QVM Execution

The molecular Hamiltonian is designed to be directly executable within QVM constraints:

- it admits static validation,
- it supports bounded operator application,
- it integrates with QPU acceleration,
- it produces auditable spectral artifacts.

No runtime modification of the Hamiltonian structure is permitted.

4.10 Summary

This section has defined the molecular Hamiltonian as a bounded, self-adjoint operator encoding all admissible molecular interactions. Its spectral properties provide a deterministic and interpretable foundation for molecular analysis.

The next section defines how operator evolution is performed on this Hamiltonian and how spectral information is extracted for computational decision-making.

5 Operator Evolution and Spectral Analysis

5.1 Overview

This section defines how molecular computation proceeds once the molecular Hamiltonian has been constructed. Rather than evolving molecular states through numerical time integration, the

QVM framework performs computation through bounded operator evolution and direct spectral analysis.

The purpose of operator evolution is not to simulate physical time, but to expose stable structures, admissible transitions, and sensitivity properties encoded in the Hamiltonian.

5.2 Discrete Operator Evolution Model

Let $\hat{H}_{\text{mol}}$ denote the molecular Hamiltonian. Operator evolution proceeds in discrete, bounded steps defined by an evolution operator:

$$\mathcal{U} = f(\hat{H}_{\text{mol}}),$$

where f is a statically declared operator function.

Admissible forms of f include:

- finite polynomials in $\hat{H}_{\text{mol}}$,
- rational functions with bounded denominators,
- spectral projection operators onto declared spectral bands.

Unbounded series, adaptive iteration, or data-dependent operator construction is prohibited.

5.3 Evolution Semantics

Given an initial molecular state $\psi_0 \in \mathcal{H}_{\text{mol}}$, evolution is defined by:

$$\psi_{n+1} = \mathcal{U}\psi_n.$$

Evolution steps:

- are deterministic,
- preserve admissibility of states,
- respect declared execution bounds,
- occur only at explicit execution boundaries.

Intermediate states are not externally observable unless explicitly permitted.

5.4 Absence of Physical Time

The evolution index n does not correspond to physical time. It represents a logical progression through operator application stages.

As a consequence:

- there is no timestep parameter,
- no numerical stability condition related to integration,
- no accumulation of discretization error.

This removes a major source of uncertainty present in classical molecular dynamics.

5.5 Spectral Decomposition

Central to the framework is the spectral decomposition of the molecular Hamiltonian:

$$\hat{H}_{\mathrm{mol}} = \sum_k \lambda_k \, \Pi_k,$$

where λ_k are eigenvalues and Π_k are the corresponding spectral projectors.

Spectral decomposition is:

- deterministic,
- reproducible,
- independent of execution backend.

Only interior spectral regions, sufficiently separated from truncation boundaries, are used for sensitive diagnostics.

5.6 Interpretation of Spectral Features

Spectral features are interpreted structurally rather than probabilistically.

In particular:

- low eigenvalues correspond to structurally stable molecular configurations,
- spectral gaps indicate robustness under perturbation,
- dense spectral regions indicate conformational flexibility,
- isolated eigenvalues correspond to distinct structural regimes.

No sampling-based interpretation is applied.

5.7 Spectral Projections and Filtering

Spectral projection operators:

$$\Pi_\Delta = \sum_{\lambda_k \in \Delta} \Pi_k$$

are used to isolate subsets of molecular configurations corresponding to declared spectral intervals Δ.

These projections enable:

- targeted analysis of stable conformations,
- elimination of high-energy or unstable states,
- controlled comparison between molecular variants.

All spectral intervals must be declared prior to execution.

5.8 Expectation Values and Observables

Observable quantities are computed as expectation values of declared operators:

$$\langle \hat{O} \rangle_\psi = \langle \psi, \hat{O}\psi \rangle.$$

Expectation values are used to derive:

- effective binding indicators,
- structural sensitivity measures,
- comparative scores between molecular candidates.

Expectation computation is deterministic and fully auditable.

5.9 Perturbation Analysis

Small structural or parameter modifications are modeled as bounded perturbations:

$$\hat{H}_{\text{mol}} \to \hat{H}_{\text{mol}} + \delta\hat{H}.$$

The resulting spectral shifts:

$$\lambda_k \to \lambda_k + \delta\lambda_k$$

provide quantitative measures of molecular sensitivity.

This forms the basis for rational lead optimization.

5.10 Execution Boundaries and Auditability

All operator evolution and spectral analysis steps:

- occur within declared execution bounds,
- produce explicit audit records,
- yield sealed and reproducible outputs.

No implicit state transitions or hidden computations are permitted.

5.11 Summary

This section has defined the deterministic operator evolution model and the role of spectral analysis in molecular computation. By replacing trajectory-based simulation with bounded operator evolution and spectral diagnostics, the framework provides global structural insight, reproducibility, and interpretability.

The next section applies these mechanisms to ligand–protein binding and molecular interaction analysis.

6 Ligand–Protein Binding and Interaction Analysis

6.1 Overview

This section applies the operator-based molecular framework to the analysis of ligand–protein interactions. Binding is treated as a structural and spectral phenomenon arising from the coupling of two molecular operator systems, rather than as a stochastic docking or trajectory-based process.

The goal is to provide a deterministic, interpretable, and auditable method for evaluating binding affinity, selectivity, and robustness.

6.2 Composite Molecular Systems

Let $\mathcal{H}_{\text{protein}}$ and $\mathcal{H}_{\text{ligand}}$ denote the state spaces of the protein and ligand, respectively. The combined system is represented by the tensor product:

$$\mathcal{H}_{\text{total}} = \mathcal{H}_{\text{protein}} \otimes \mathcal{H}_{\text{ligand}}.$$

No implicit interaction is assumed at the level of state representation. All coupling is introduced explicitly through interaction operators.

6.3 Uncoupled Hamiltonian Structure

Prior to interaction, the uncoupled Hamiltonian of the composite system is defined as:

$$\hat{H}_0 = \hat{H}_{\text{protein}} \otimes I + I \otimes \hat{H}_{\text{ligand}}.$$

The spectral structure of $\hat{H}_0$ reflects independent conformational regimes of the protein and ligand.

6.4 Interaction Hamiltonian

Ligand–protein interaction is introduced through an explicit interaction Hamiltonian:

$$\hat{H}_{\text{int}} = \sum_{(i,j)} \hat{I}_{ij},$$

where $\hat{I}_{ij}$ denotes bounded interaction operators coupling protein site i with ligand site j.

Interaction operators encode:

- steric complementarity,
- electrostatic coupling,
- hydrogen bonding potential,
- hydrophobic interaction effects.

All interaction terms are explicitly declared and bounded.

6.5 Total Coupled Hamiltonian

The full Hamiltonian of the interacting system is given by:

$$\hat{H}_{\text{total}} = \hat{H}_0 + \hat{H}_{\text{int}}.$$

This operator fully determines admissible bound and unbound configurations within the QVM framework.

6.6 Spectral Signature of Binding

Binding is identified through changes in the spectral structure of the Hamiltonian upon introduction of $\hat{H}_{\text{int}}$.

Key indicators include:

- emergence of low-lying eigenvalues not present in $\hat{H}_0$,
- spectral gap formation indicating stable binding modes,
- localization of eigenstates in interaction-relevant subspaces.

No probabilistic docking scores are used.

6.7 Binding Affinity Measures

Binding affinity is quantified through deterministic spectral metrics, including:

- depth of spectral stabilization relative to the uncoupled system,
- robustness of low-energy eigenstates under perturbation,
- absence of competing unstable interaction modes.

These measures provide a total ordering of ligand candidates without reliance on stochastic sampling.

6.8 Selectivity and Off-Target Analysis

Selectivity is evaluated by comparing spectral responses of a ligand across multiple protein targets.

Off-target risk is indicated by:

- unintended low-energy eigenstates in non-target systems,
- spectral similarity across unrelated protein environments,
- lack of strong spectral gaps for the intended target.

This enables early identification of promiscuous or unsafe candidates.

6.9 Conformational Flexibility and Induced Fit

Conformational adaptation is represented through spectral broadening and mode coupling rather than explicit trajectory deformation.

Induced-fit behavior corresponds to:

- smooth spectral transitions under interaction coupling,
- preservation of boundedness and self-adjointness,
- absence of discontinuous state transitions.

This avoids the need for explicit conformational sampling.

6.10 Ranking and Decision Criteria

Ligand candidates are ranked using a composite of spectral criteria:

- binding spectral depth,
- stability under perturbation,

- selectivity margin relative to off-target systems.

All ranking decisions are reproducible and traceable to explicit operator properties.

6.11 Summary

This section has defined ligand–protein binding as a deterministic operator coupling problem and has shown how binding affinity and selectivity emerge from spectral analysis of the coupled Hamiltonian.

The next section maps these computations onto QVM and QPU execution and discusses practical deployment considerations.

7 QVM/QPU Execution and Architecture Mapping

7.1 Overview

This section describes how the molecular operator framework defined in the preceding sections is executed within the QVM architecture and accelerated through QPU backends. The focus is on deterministic execution, bounded resource usage, and full auditability, rather than raw throughput or heuristic optimization.

The execution model is designed to ensure that molecular computations remain reproducible, verifiable, and independent of specific hardware implementations.

7.2 QVM Execution Model

The Quantum Virtual Machine (QVM) provides a governed execution environment for operator-based computation. Within this environment, molecular computation is expressed as a sequence of explicitly declared operator applications.

Key properties of QVM execution include:

- deterministic scheduling of operator application,
- immutability of input states during execution,
- explicit declaration of execution boundaries,
- prohibition of implicit control flow or data-dependent branching.

All molecular computations are executed as closed programs with statically known structure.

7.3 Operator Programs

A molecular computation is encoded as an operator program consisting of:

- a declared molecular state space,
- a validated molecular Hamiltonian,
- a finite set of evolution and projection operators,
- a set of declared observables and spectral queries.

Operator programs are subject to static validation prior to execution. Programs failing validation are rejected without partial execution.

7.4 Execution Phases

Execution proceeds in well-defined phases:

1. state initialization and validation,
2. Hamiltonian instantiation and verification,
3. operator evolution execution,
4. spectral decomposition and projection,
5. observable evaluation and result sealing.

No phase may alter the structure of subsequent phases.

7.5 QPU Acceleration Model

The Quantum Processing Unit (QPU) acts as an acceleration backend for specific operator classes. QPUs are not required to be quantum hardware; they may include classical accelerators, vectorized units, or specialized spectral engines.

QPU responsibilities include:

- efficient application of bounded operators,
- parallel evaluation of operator blocks,
- acceleration of spectral decomposition routines.

The QVM defines the execution semantics; the QPU implements approved kernels under strict constraints.

7.6 Operator-to-Kernel Mapping

Each operator class is mapped to one or more execution kernels. This mapping is:

- explicit and versioned,
- statically analyzable,
- independent of data-dependent behavior.

Kernel execution may be parallelized internally, but the logical execution order observed by the QVM remains deterministic.

7.7 Backend Independence

QVM programs are backend-agnostic. Identical operator programs executed on different QPU backends must produce identical results up to declared numerical tolerances.

Backend differences are constrained to:

- internal parallelization strategies,
- low-level numerical optimizations,
- hardware-specific instruction scheduling.

No backend is permitted to alter operator semantics or execution order.

7.8 Numerical Precision and Stability

Numerical precision is declared explicitly as part of the execution configuration. All kernels must respect declared precision bounds.

Stability guarantees include:

- bounded numerical error propagation,
- reproducible rounding behavior,
- deterministic handling of degeneracies.

Adaptive precision adjustment during execution is not permitted.

7.9 Audit Trails and Result Sealing

Every execution produces a complete audit trail containing:

- operator program identifiers,
- Hamiltonian hashes,
- kernel version identifiers,
- spectral and observable outputs.

Results are sealed upon completion and may be independently verified without re-execution.

7.10 Failure Modes and Safety Guarantees

Execution failures result in:

- immediate termination,
- explicit error signaling,
- no partial or ambiguous outputs.

Undefined behavior, silent fallback, or heuristic recovery is prohibited.

7.11 Summary

This section has defined how molecular operator programs are executed within the QVM architecture and accelerated through QPU backends. The separation of execution semantics from hardware implementation ensures determinism, reproducibility, and long-term stability.

The next section discusses computational guarantees, limitations, and the scope of applicability of the framework.

8 Computational Guarantees, Limitations, and Scope

8.1 Overview

This section delineates the formal guarantees provided by the QVM-based molecular computation framework, as well as its inherent limitations and intended scope of applicability. The goal is to

establish clear expectations regarding correctness, stability, and interpretability, while explicitly avoiding overextension beyond the framework's design assumptions.

The guarantees described here arise from architectural and mathematical constraints rather than empirical performance claims.

8.2 Determinism Guarantees

All computations performed within the framework are deterministic by construction. Given identical inputs, including:

- molecular state space declarations,
- operator definitions and parameters,
- execution configuration and precision,
- QVM program structure,

the system produces identical outputs across executions and hardware backends.

No source of randomness, implicit sampling, or nondeterministic scheduling is permitted at any layer of execution.

8.3 Reproducibility and Verifiability

Results produced by the system are reproducible over time and independently verifiable. Reproducibility is ensured through:

- explicit operator and Hamiltonian declarations,
- versioned kernel mappings,
- sealed execution artifacts,
- immutable audit records.

Independent verification may be performed without access to the original execution environment, provided the declared execution context is available.

8.4 Boundedness and Resource Guarantees

All computations are statically bounded. Prior to execution, upper bounds are established for:

- state space dimension,
- operator count and composition depth,
- spectral resolution parameters,
- numerical precision requirements.

Programs exceeding declared bounds are rejected at validation time. No dynamic resource escalation is allowed during execution.

8.5 Numerical Stability

Numerical stability is guaranteed within declared precision bounds. Stability properties include:

- bounded propagation of numerical error,
- consistent handling of near-degenerate spectra,
- deterministic resolution of operator ordering.

Numerical artifacts arising from truncation or discretization are explicitly documented and do not accumulate across executions.

8.6 Interpretability Guarantees

All outputs are interpretable in terms of declared operator semantics. Spectral results correspond directly to properties of the constructed Hamiltonian and its constituents.

The framework guarantees that:

- no hidden state or implicit inference is involved,
- all observables map to explicit operators,
- all ranking and decision criteria are traceable.

Interpretability is a structural property of the system, not a post hoc explanation layer.

8.7 Limitations of the Framework

The framework does not claim to:

- simulate real-time molecular dynamics,
- replace experimental validation,
- model biological systems beyond molecular interaction,
- infer emergent biological behavior.

The representation is limited by the chosen discretization and operator library. Phenomena not expressible within this operator set are outside the scope of the framework.

8.8 Approximation and Modeling Assumptions

All results depend on modeling assumptions encoded in operator definitions and discretization choices. These assumptions are explicit and form part of the audit record.

The framework guarantees correctness relative to its declared model, not absolute physical truth.

8.9 Scope of Applicability

The framework is intended for:

- comparative analysis of molecular candidates,
- rational lead optimization,
- early-stage binding and selectivity assessment,
- risk reduction prior to experimental screening.

It is not intended for late-stage clinical prediction or organism-level modeling.

8.10 Integration with Broader Pipelines

Outputs produced by the framework are designed to integrate with:

- experimental screening workflows,
- downstream statistical and biological analysis,
- regulatory documentation processes.

The system acts as a deterministic computational instrument within a larger discovery pipeline.

8.11 Summary

This section has established the formal guarantees, limitations, and scope of the QVM-based molecular computation framework. By clearly defining what the system does and does not claim to do, it provides a stable and trustworthy foundation for scientific and industrial use.

With this, the core technical specification of the QVM pharmacology whitepaper is complete.