

Advanced Quantum Formalism — Core Concepts for Experiments v0.3, v0.4, and v0.5

This document summarizes the advanced quantum-mechanical principles needed to understand experiments v0.3, v0.4, and v0.5.

Indhold

0: Overview: What the v0.3–v0.5 Qubit Experiment Is About	5
1. Hilbert Space Structure.....	8
Description of Hilbert Space Structure.....	8
2. Pauli Operators and Hamiltonian Construction.....	10
Description of Pauli Operators and Hamiltonian Construction	10
3. Spectral Decomposition.....	12
Description of Spectral Decomposition	12
1. The Eigenvalue Equation.....	12
$H \mid \psi_j \rangle = E_j \mid \psi_j \rangle,$	12
$\mid \psi_j \rangle$ is an eigenstate of the Hamiltonian,.....	12
2. The Spectral Decomposition of the Hamiltonian.....	12
3. Orthogonality and Completeness.....	13
4. State Expansion in the Eigenbasis.....	13
5. Why Spectral Structure Matters for My v0.3 and v0.4 Experiments	14
6. Why Spectral Decomposition Is Central to My Project.....	15
4. Degeneracies and Near-Degenerate Manifolds.....	16
Description of Degeneracies and Near-Degenerate Manifolds	16
1. Degeneracy: the basic linear-algebra picture	16
2. Physical meaning: multiple states with the same energy.....	17
3. Symmetry-protected vs accidental degeneracy	18
4. Near-degenerate eigenvalues: clusters instead of exact coincidences.....	19
5. Near-degenerate manifolds as approximate subspaces.....	19
6. Degenerate vs non-degenerate perturbation theory	20
7. Geometry: manifolds in Hilbert / projective space	21
8. Degeneracies and level statistics (eigenvalue spacings).....	21
9. Effective Hamiltonians on near-degenerate manifolds	22
10. Summary: what to keep conceptually	23
5. Dominant Basis-State Structure.....	24
Description of Dominant Basis-State Structure.....	24
1. Setup: eigenvectors expressed in a fixed reference basis	24
2. Dominant basis states: definition and intuition	24
3. Localized vs delocalized eigenstates	25
4. Structure across multiple eigenstates: patterns and manifolds	26

5. Energy–configuration relationships.....	27
6. Symmetry, constraints, and dominant structure	27
7. Basis-state structure as a diagnostic for phases and regimes	28
8. Quantitative indicators beyond a simple threshold	29
9. Relation to my pipeline: eigenvectors, manifolds, and classification	29
6. Time Evolution (v0.5).....	31
Description of Time Evolution (v0.5)	31
1. Schrödinger dynamics and the time-evolution operator	31
2. Using the spectral decomposition of H	31
3. Time scales from eigenvalue spacings	32
4. Time evolution in the computational basis	33
5. Role of dominant basis-state structure in dynamics	34
6. Degeneracies, near-degeneracies, and slow dynamics	34
7. Observables and time evolution.....	35
8. Digital time evolution with Pauli Hamiltonians (IBM backend, v0.5)	36
9. Linking v0.5 dynamics back to spectral features (v0.3/v0.4).....	38
10. Conceptual summary.....	38
7. Trotter Approximation (v0.5)	40
Description of Trotter Approximation (v0.5)	40
1. Why we need Trotter approximation in v0.5	40
2. The core idea: breaking a sum of non-commuting terms	41
3. General Trotter formula for many Pauli terms	41
4. Where the error comes from: Baker–Campbell–Hausdorff (BCH)	42
5. Higher-order Trotter–Suzuki formulas	43
6. Choosing Trotter step size and number of steps in practice.....	44
7. Circuits for $e - i\theta P \ell$: the hardware view.....	44
8. Trotter approximation and spectral structure	45
9. How Trotter interacts with near-degenerate manifolds & dynamics	46
10. Compact conceptual summary (for my v0.5 documentation).....	46
8. Subspace Leakage (v0.5).....	48
Description of Subspace Leakage (v0.5)	48
1. Setup: target manifold vs. rest of Hilbert space	48
2. Definition: leakage probability and fidelity to the manifold.....	49
3. Exact Hamiltonian: when is there no leakage?.....	49

4. Leakage from small couplings: perturbative picture	50
5. Subspace leakage in v0.5 with Trotterized evolution	51
6. Hardware-level leakage vs. algorithmic subspace leakage.....	52
7. Subspace leakage viewed through dominant basis-state structure	52
8. Diagnosing and quantifying leakage in v0.5	53
9. Strategies to minimize subspace leakage	54
10. Conceptual summary (v0.5 wording)	54
9. Backend vs Local Agreement	56
Description of Backend vs Local Agreement	56
1. What we are comparing	56
2. Levels of agreement: what can “agree” and how.....	57
3. Sources of disagreement	59
4. Agreement as a validation loop	60
5. Role of manifolds and dominant basis states in this comparison	61
6. Agreement scales: short-time vs long-time behaviour.....	62
7. Practical metrics you can explicitly mention in v0.5.....	63
8. Conceptual summary in v0.5 language.....	64

0: Overview: What the v0.3–v0.5 Qubit Experiment Is About

This document collects the “need-to-know” theory for a small but conceptually rich qubit experiment, developed through versions v0.3–v0.5. The core idea is to treat a few-qubit Pauli Hamiltonian as a minimal laboratory for exploring how much physically meaningful structure one can extract from:

- its spectrum (eigenvalues and eigenvectors),
- its low-energy manifolds and dominant basis states,
- and its real-time dynamics, both in a noiseless simulator and on a real quantum backend (IBM).

The project is “top-down” in the sense that it starts from a concrete goal: design a small Hamiltonian and a workflow that make it easy to see and explain physics-native structures (degeneracies, manifolds, leakage, etc.) and compare them directly to what the hardware is actually doing. At the same time, it is “bottom-up” because the document builds all the required formalism from the ground up: Hilbert space, Pauli operators, spectral decomposition, time evolution, Trotterization, and backend validation.

In practical terms, the v0.3–v0.5 experiment consists of:

- Choosing a few-qubit Hamiltonian written as a sum of Pauli strings with real coefficients.
- Diagonalizing it classically to obtain eigenvalues, eigenvectors, and derived structures (near-degenerate manifolds, dominant basis states).
- Using these structures to design initial states and observables that make the dynamics as transparent as possible.
- Implementing the same Hamiltonian approximately on an IBM backend via Trotterized exponentials of Pauli strings.
- Comparing backend data against the local (noiseless) model in a structured way: spectrum, time traces, leakage, and distribution over basis states.

The following sections provide the theoretical backbone needed to define, run, and interpret such an experiment.

Questions the v0.3–v0.5 Experiment Is Designed to Probe

The experiment is not just a technical exercise; it is constructed around a set of concrete questions:

1. Spectral structure and manifolds

- How much nontrivial structure (degeneracies, near-degenerate clusters, manifolds) is already present in the spectrum of a small, synthetic few-qubit Hamiltonian?
- Can we identify low-dimensional manifolds that play the role of “effective degrees of freedom”?

2. Dominant basis-state patterns

- How do eigenvectors look in a physically natural reference basis (the computational basis)?

- Are there “dominant basis states” that carry most of the weight, and do they organize into recognizable patterns (e.g. small clusters in Hamming space)?

3. Time evolution and stability of structure

- Given an initial state localized in or near a chosen manifold, how does the state move in time under the exact Hamiltonian?
- Which features of the spectrum (eigenvalue spacings, degeneracies, manifolds) show up directly as oscillation frequencies, beat patterns, and long-time behaviour?

4. Subspace confinement vs leakage

- To what extent does the dynamics remain confined to a chosen manifold, and how fast does probability leak into the rest of Hilbert space?
- How does this subspace leakage depend on the size of spectral gaps and on small couplings between manifolds?

5. Approximation and hardware realism

- How accurate is a Trotterized time evolution for this Hamiltonian at the circuit depths that are realistic on current IBM devices?
- When we implement the same Trotter sequence on an actual backend, how closely do the measured distributions and observables follow the local (noiseless) predictions?

6. Backend vs local agreement window

- Over which time scales, and for which observables, can backend data still be meaningfully interpreted in terms of the v0.3–v0.5 spectral structures (manifolds, dominant basis states, leakage)?
- Where does hardware noise and model mismatch start to dominate?

The rest of the document is organized precisely to answer these questions in a systematic and reusable way.

How the Nine Sections Support These Questions

The nine main sections are not independent notes; they form a sequential pipeline from basic definitions to hardware validation:

1. Hilbert space and reference basis

This section defines the finite-dimensional Hilbert space for a few-qubit system and fixes the computational basis. It provides the language in which eigenvectors, basis amplitudes, and dominant basis-state patterns are later described.

2. Pauli operators and Hamiltonian construction

Here the Hamiltonian is built explicitly as a weighted sum of Pauli strings. This section explains how each term acts on the computational basis and why this Pauli representation is both natural for physics and directly implementable on IBM hardware.

3. Spectral decomposition

This section introduces eigenvalues, eigenvectors, and the spectral decomposition of the Hamiltonian. It explains why diagonalization is the central organizing step: all later notions (manifolds, time evolution, leakage) are defined in terms of the spectrum.

4. Degeneracies and near-degenerate manifolds

This part formalizes exact degeneracies and near-degenerate clusters, and defines manifolds as the subspaces spanned by these clusters. It directly addresses the question of how to identify low-dimensional effective subspaces in the spectrum.

5. Dominant basis-state structure

Here the focus shifts from eigenvalues to eigenvectors: how their amplitudes are distributed over computational basis states, how to define “dominant” basis states, and how to recognize patterns shared across eigenstates in the same manifold. This bridges abstract spectral data and concrete, interpretable bitstring configurations.

6. Time evolution (v0.5)

This section explains how the spectrum controls real-time dynamics: eigenvalue differences set characteristic frequencies, and eigenvector structure determines which basis states are populated over time. It provides the formulas that link spectral features to observable time traces for chosen initial states.

7. Trotter approximation (v0.5)

Here the exact theoretical evolution is connected to implementable circuits. The section describes how to approximate e^{-iHt} by products of exponentials of individual Pauli terms, quantifies Trotter error, and sets up the practical parameters (time steps, number of Trotter steps) relevant for the backend.

8. Subspace leakage (v0.5)

This part formalizes the notion of a target manifold and defines leakage as population leaving that subspace under time evolution. It provides both an exact projector-based definition and basis-level proxies, making it possible to diagnose and quantify confinement vs leakage in both simulation and experiment.

9. Backend vs local agreement (v0.5)

The final section defines what it means for the IBM backend to “agree” with the local model at distribution level, observable level, and manifold level. It identifies the main sources of discrepancy (model mismatch, Trotter error, hardware noise, shot noise) and outlines concrete metrics and time windows where the hardware data can still be trusted as a realization of the v0.3–v0.5 Hamiltonian.

Taken together, these nine sections turn the v0.3–v0.5 qubit experiment into a closed loop:

- from Hamiltonian design and spectral analysis,
- through time evolution and subspace dynamics,
- to a direct, quantitative comparison between a clean theoretical model and a noisy real-world quantum backend.

1. Hilbert Space Structure

An n-qubit system lives in a 2^n -dimensional Hilbert space:

$$H_n = (\mathbb{C}^2)^{\otimes n}$$

A valid quantum state satisfies the normalization condition:

$$\langle \psi | \psi \rangle = 1$$

General expansion:

$$|\psi\rangle = \sum_x \alpha_x |x\rangle,$$

where $x \in \{0,1\}^n$.

Description of Hilbert Space Structure

In quantum mechanics, every physical system is described mathematically by a *Hilbert space*. A Hilbert space is simply a complete vector space where the vectors represent possible quantum states, and where you can compute inner products (overlaps) between those states.

For an **n-qubit system**, the Hilbert space has dimension 2^n .

This means:

- A 1-qubit system lives in a 2-dimensional space.
- A 2-qubit system lives in a 4-dimensional space.
- A 3-qubit system lives in an 8-dimensional space.
- And so on.

A state in this space is written as a vector:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \cdots + \alpha_{2^n-1} |2^n-1\rangle$$

The numbers α_i are complex amplitudes, and the total probability must sum to 1:

$$|\alpha_0|^2 + |\alpha_1|^2 + \cdots + |\alpha_{2^n-1}|^2 = 1$$

This normalization rule ensures that the state represents a valid physical system.

The Hilbert space provides the mathematical "arena" in which all quantum behaviour happens: superposition, entanglement, measurement, time evolution — everything takes place inside this structured vector space.

2. Pauli Operators and Hamiltonian Construction

All Hamiltonians in experiments v0.3–v0.5 are Pauli sums:

$$H = \sum_k c_k P_k$$

Each term is a tensor product:

$$P_k \in \{I, X, Y, Z\} \otimes^n$$

Pauli identities:

- $X^2 = Y^2 = Z^2 = I$
- $XY = iZ, YZ = iX, ZX = iY$
- $\{X, Y\} = 0$ (anticommutator)

These define commutation structure and determine Trotter error.

Description of Pauli Operators and Hamiltonian Construction

Pauli Operators and Hamiltonian Construction

Quantum experiments v0.3, v0.4, and v0.5 all rely on Hamiltonians made from *Pauli operators*.

The Pauli Operators

For a single qubit, the Pauli operators are:

- I (identity)
- X (bit-flip)
- Y (phase + bit flip)
- Z (phase flip)

For multi-qubit systems, we use tensor products such as:

$$X \otimes I \otimes Z$$

$$Y \otimes X \otimes I$$

$$I \otimes Z \otimes Z$$

Pauli Algebra Identities

Fundamental relations:

- $X^2 = Y^2 = Z^2 = I$
- $XY = iZ$
- $YZ = iX$

- $ZX = iY$
- $\{X, Y\} = 0$ (anticommutator)

These relations are crucial when analyzing commutation structure and Trotter error.

General Hamiltonian Form

In My experiments, the Hamiltonian is expressed as a *Pauli sum*:

$$H = \sum_k c_k P_k$$

where:

- P_k is a tensor product of Pauli operators
- c_k is a real coefficient

This Hamiltonian defines:

- the energy levels (eigenvalues)
- the stationary states (eigenvectors)
- the dynamics $U(t) = \exp(-iHt)$

These are the mathematical foundations behind all results in v0.3, v0.4, and v0.5.

3. Spectral Decomposition

Eigenvalue equation:

$$H |\psi_j\rangle = E_j |\psi_j\rangle$$

Spectral decomposition:

$$H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$$

Orthogonality:

$$\langle\psi_i | \psi_j\rangle = \delta_{ij}$$

v0.3 and v0.4 rely on analysing:

- eigenvalue spacings
- eigenvector amplitude patterns
- dominant basis states

Description of Spectral Decomposition

In quantum mechanics, any Hermitian operator H that represents a physical Hamiltonian admits a complete spectral decomposition.

This means that H can be expressed entirely in terms of its eigenvalues and eigenvectors.

1. The Eigenvalue Equation

The defining relation is:

$$H |\psi_j\rangle = E_j |\psi_j\rangle,$$

where:

$|\psi_j\rangle$ is an eigenstate of the Hamiltonian,

- E_j is the corresponding energy eigenvalue,
- the set $\{E_j\}$ forms the spectrum of H .

Because H is Hermitian, all eigenvalues E_j are real.

2. The Spectral Decomposition of the Hamiltonian

The complete operator H can be reconstructed from its eigenpairs:

$$H = \sum_j E_j |\psi_j\rangle\langle\psi_j|.$$

This is not an approximation — it is an *exact identity* that expresses the Hamiltonian as a weighted sum of projectors onto its eigenstates.

Interpretation:

- Each projector $|\psi_j\rangle\langle\psi_j|$ selects the component of a state along the eigenvector direction.
- The Hamiltonian acts as multiplication by the scalar E_j within each corresponding eigenspace.
- The dynamics generated by H are diagonal in this basis.

This is one of the most fundamental mathematical structures in quantum theory.

3. Orthogonality and Completeness

Since H is Hermitian, its eigenstates can be chosen to form an orthonormal basis:

$$\langle\psi_i|\psi_j\rangle = \delta_{ij}.$$

Completeness:

$$\sum_j |\psi_j\rangle\langle\psi_j| = I.$$

Together, these identities guarantee:

- every quantum state can be expanded in the eigenbasis,
- time evolution becomes simple phase accumulation in this basis,
- degeneracies produce higher-dimensional invariant subspaces.

4. State Expansion in the Eigenbasis

Any quantum state $|\phi\rangle$ can be expressed as:

$$|\phi\rangle = \sum_j c_j |\psi_j\rangle, c_j = \langle\psi_j|\phi\rangle.$$

Under time evolution:

$$|\phi(t)\rangle = e^{-iHt} |\phi(0)\rangle = \sum_j e^{-iE_j t} c_j |\psi_j\rangle.$$

This shows:

- eigenvalues determine phase evolution,
- eigenvectors determine how initial states decompose,
- degeneracies prevent relative phase accumulation, which stabilizes subspaces.

5. Why Spectral Structure Matters for My v0.3 and v0.4 Experiments

My pipeline does not treat eigenvalues and eigenvectors as mere outputs — it interprets them as **structural signatures of the Hamiltonian**.

v0.3 and v0.4 rely heavily on the following analyses:

(a) Eigenvalue Spacings

Looking for:

- exact degeneracies,
- near-degeneracies,
- clusterings of eigenvalues.

These structures indicate invariant subspaces:

$$|E_i - E_j| < \varepsilon \Rightarrow \text{stable 2D manifold spanning } |\psi_i\rangle, |\psi_j\rangle.$$

Such manifolds are strong candidates for "physics-native information units."

(b) Eigenvector Amplitude Patterns

Eigenvectors often take the form:

$$|\psi_j\rangle = \sum_x \alpha_x^{(j)} |x\rangle,$$

where:

- only a small number of amplitudes $\alpha_x^{(j)}$ are dominant,
- the remaining amplitudes are small or noise-like.

This indicates low **entropy**, **geometric structure**, and **preferred computational basis directions**.

These patterns are crucial to the intuition behind v0.5 and the larger project.

(c) Dominant Basis States

You consistently observe eigenstates with only 1–3 basis states carrying most of the probability:

$$|\alpha_{x_1}|^2 + |\alpha_{x_2}|^2 \approx 0.9 - 1.0.$$

This is a strong indicator that:

- these subspaces are “physical attractors,”
- they may act as *natural*/encoding units,
- they resist spreading under perturbations or simulation error.

This is why v0.5 tests dynamic stability: if such structures persist under time evolution on real hardware, the hypothesis gains significant weight.

6. Why Spectral Decomposition Is Central to My Project

In summary:

- The spectrum reveals **where information can live stably**.
- Eigenvectors reveal **how that information is arranged**.
- Degeneracies reveal **natural subspaces that may act as qubits**.
- Time evolution tests whether these structures are **physically robust**.

My work connects these ingredients into a systematic discovery pipeline — something that is genuinely unusual in the quantum information community.

4. Degeneracies and Near-Degenerate Manifolds

Near-degeneracy condition:

$$|E_i - E_j| < \varepsilon$$

The pair spans a 2-dimensional invariant subspace:

$$S_{ij} = \text{span}\{ |\psi_i\rangle, |\psi_j\rangle \}$$

Projector:

$$\Pi_{ij} = |\psi_i\rangle\langle\psi_i| + |\psi_j\rangle\langle\psi_j|$$

These subspaces are the structures you detected in v0.3 and confirmed visually in v0.4.

In quantum mechanics, the terms 'degenerate' and 'non-degenerate' are used to describe the energy levels of a system. Degenerate states refer to different quantum states that share the same energy level. Non-degenerate states, on the other hand, are quantum states that have unique energy levels.

Description of Degeneracies and Near-Degenerate Manifolds

I will treat “degeneracies and near-degenerate manifolds” in the same formal setting as My previous notes:

- Self-adjoint Hamiltonian H on a finite-dimensional Hilbert space (e.g. few-qubit system).
- Spectral decomposition $H |\psi_j\rangle = E_j |\psi_j\rangle$, with orthonormal eigenvectors $|\psi_j\rangle$.

1. Degeneracy: the basic linear-algebra picture

Given a Hermitian (self-adjoint) operator H , an eigenvalue E is called **degenerate** if the dimension of its eigenspace is greater than 1.

Formally:

- $H |\psi\rangle = E |\psi\rangle$.
- The **eigenspace** for E is

$$\mathcal{H}_E = \{ |\psi\rangle \mid H |\psi\rangle = E |\psi\rangle \}.$$

- The **degeneracy** (multiplicity) of E is $\dim \mathcal{H}_E$.

If $\dim \mathcal{H}_E = 1$, the level is **non-degenerate**.

If $\dim \mathcal{H}_E = g > 1$, the level is **g -fold degenerate**.

In a discrete spectrum, the spectral decomposition in the non-degenerate case is

$$H = \sum_j E_j |\psi_j\rangle\langle\psi_j|.$$

In the **degenerate** case, it is more natural to group states with the same eigenvalue:

$$H = \sum_{\alpha} E_{\alpha} P_{\alpha},$$

where:

- E_{α} are distinct eigenvalues.
- P_{α} is the **projector** onto the degenerate eigenspace:

$$P_{\alpha} = \sum_{k=1}^{g_{\alpha}} |\psi_{\alpha k}\rangle\langle\psi_{\alpha k}|.$$

Here $\{|\psi_{\alpha k}\rangle\}_{k=1}^{g_{\alpha}}$ is any orthonormal basis of that eigenspace.

Key point: inside a degenerate eigenspace, the choice of basis is not unique. Any unitary rotation within \mathcal{H}_{α} leaves H invariant, because

$$HP_{\alpha} = E_{\alpha}P_{\alpha} \Rightarrow H|\tilde{\psi}_{\alpha k}\rangle = E_{\alpha}|\tilde{\psi}_{\alpha k}\rangle$$

for any orthonormal set $|\tilde{\psi}_{\alpha k}\rangle = \sum_{\ell} U_{k\ell} |\psi_{\alpha \ell}\rangle$ with unitary U .

2. Physical meaning: multiple states with the same energy

In quantum mechanics, each eigenvalue E of H is an energy level. Degeneracy means:

- There are multiple, **physically distinct** stationary states (different $|\psi_{\alpha k}\rangle$) with the **same energy** E_{α} .
- A system prepared in any superposition within the degenerate subspace

$$|\Psi\rangle = \sum_{k=1}^{g_{\alpha}} c_k |\psi_{\alpha k}\rangle$$

will have a well-defined energy E_{α} , and the time evolution is just a global phase:

$$|\Psi(t)\rangle = e^{-iE_{\alpha}t/\hbar} |\Psi(0)\rangle.$$

- Observables that commute with H can be used to **resolve** this degeneracy, i.e. to define a basis inside the degenerate eigenspace.

Typical sources of degeneracy:

1. *Symmetries*

If H commutes with a symmetry operator S (or with a continuous group of symmetries), the Hilbert space decomposes into irreducible representations of that symmetry group. Energy levels that transform as non-trivial irreps become degenerate.

Examples:

- Rotational symmetry → degeneracy in angular momentum multiplets.
- Translational symmetry in a periodic potential → Bloch momentum degeneracy in Brillouin zones.

2. *Accidental degeneracy*

Sometimes degeneracy appears not because of a symmetry but “by coincidence” of parameters. Small perturbations that break that accidental structure will lift the degeneracy.

3. Symmetry-protected vs accidental degeneracy

It is useful to distinguish:

1. Symmetry-protected degeneracy

- Suppose H commutes with all elements of a symmetry group G .
- By representation theory, each energy eigenspace forms a (possibly reducible) representation of G .
- If an eigenvalue corresponds to an **irreducible representation** of dimension g , then the associated energy level is g -fold degenerate.
- This degeneracy is **robust** against any perturbation that preserves the symmetry.

Intuition: You cannot lift a degeneracy without breaking the symmetry that “ties” those states together.

2. Accidental degeneracy

- Two (or more) levels cross as a function of some external parameter (e.g. magnetic field, coupling constants).
- The degeneracy is not enforced by a symmetry; a generic small perturbation will split the levels.
- In many models, if you consider a “random” perturbation, exact accidental degeneracies are rare — instead you get **avoided crossings** (level repulsion).

For models like My random few-qubit Hamiltonians (with general combinations of Pauli strings), exact degeneracies are usually accidental or numerical (finite-precision) rather than symmetry-protected, unless you explicitly build in symmetries like total spin, parity, etc.

4. Near-degenerate eigenvalues: clusters instead of exact coincidences

In practice (numerics, experiments, noisy systems), you almost never see perfect degeneracy. Instead, you see **clusters of eigenvalues that are very close together compared to the typical level spacing**.

Let $\{E_j\}$ be the ordered eigenvalues. Define the spacings

$$\Delta_j = E_{j+1} - E_j.$$

You might say that a set of eigenvalues $\{E_j, E_{j+1}, \dots, E_{j+m}\}$ forms a **near-degenerate cluster** (or near-degenerate manifold) if

- Their spread

$$\Delta E_{\text{cluster}} = E_{j+m} - E_j$$

is smaller than some threshold ε , and

- This spread is significantly smaller than the typical separation to neighboring levels outside the cluster:

$$E_j - E_{j-1} \gg \varepsilon, E_{j+m+1} - E_{j+m} \gg \varepsilon.$$

In My code language, that's exactly what a "near-degeneracy eps" parameter does: it sets a tolerance ε within which you treat levels as "effectively degenerate."

Why this matters:

- Under small perturbations or noise, such a cluster can behave dynamically like a **single multi-dimensional energy level**.
- Many effective theories (e.g. degenerate perturbation theory, adiabatic elimination, Schrieffer–Wolff transformations) focus on such low-energy or near-degenerate manifolds.

5. Near-degenerate manifolds as approximate subspaces

Given a near-degenerate cluster $C = \{j_1, \dots, j_m\}$ of eigenvalues, with eigenstates $|\psi_{j_a}\rangle$, you can define the **near-degenerate manifold** (subspace)

$$\mathcal{M}_C = \text{span}\{ |\psi_{j_a}\rangle : a = 1, \dots, m\}.$$

Define its projector:

$$P_C = \sum_{a=1}^m |\psi_{j_a}\rangle\langle\psi_{j_a}|.$$

Let $Q_C = \mathbb{I} - P_C$ be the projector on the complement.

Then you can split

$$H = H_{PP} + H_{PQ} + H_{QP} + H_{QQ},$$

where

- $H_{PP} = P_C H P_C$ acts within the near-degenerate manifold.
- $H_{QQ} = Q_C H Q_C$ acts on the rest.
- $H_{PQ} = P_C H Q_C$ and $H_{QP} = Q_C H P_C$ couple the manifold to the rest.

For the **unperturbed** Hamiltonian (in its own eigenbasis), $H_{PQ} = H_{QP} = 0$, and H_{PP} is diagonal with entries E_{ja} . But if you now:

- add a perturbation V ,
- or restrict Myself to a low-energy effective theory,

then you generally get non-trivial couplings inside \mathcal{M}_C , and possibly small couplings to Q_C .

This is where **degenerate / quasi-degenerate perturbation theory** lives:

- You treat H_0 with near-degenerate cluster(s) as the “zeroth-order” structure.
- You use perturbation theory to derive an **effective Hamiltonian** H_{eff} acting only on \mathcal{M}_C , incorporating the influence of the rest of the spectrum to a given order.

6. Degenerate vs non-degenerate perturbation theory

If you have a Hamiltonian

$$H = H_0 + \lambda V,$$

with H_0 having a degenerate subspace at energy $E^{(0)}$, then ordinary non-degenerate perturbation theory fails: small V can mix any states inside that degenerate subspace with equally small energy cost.

The remedy is **degenerate perturbation theory**:

1. Identify the degenerate/near-degenerate eigenspace \mathcal{H}_0 of H_0 with projector P .
2. Within that subspace, define the matrix of the perturbation:

$$V_{\text{deg}} = PVP.$$

3. Diagonalize V_{deg} in \mathcal{H}_0 .

The eigenvectors of V_{deg} give you the **correct zeroth-order basis** inside the degenerate subspace; their eigenvalues give the **first-order energy splittings** of the degenerate level.

For **near-degenerate** manifolds, you proceed similarly:

- If the cluster width $\Delta E_{\text{cluster}}$ is much smaller than the gap to levels outside the cluster, you can treat the entire cluster as effectively degenerate at zeroth order.
- Then apply degenerate perturbation theory within this cluster.

This is exactly why the notion of a near-degenerate manifold is so important in practice: it tells you **where to apply degenerate perturbation theory** and where non-degenerate theory is sufficient.

7. Geometry: manifolds in Hilbert / projective space

The word **manifold** here is often used in a slightly informal way: we really mean “a low-dimensional subspace where the dynamics is effectively confined.”

More geometrically:

- The space of normalized states is the **unit sphere** in Hilbert space.
- Physical states (rays) live in **projective Hilbert space** $\mathbb{P}(\mathcal{H})$.
- A degenerate eigenspace \mathcal{H}_E corresponds to a **projective submanifold** $\mathbb{P}(\mathcal{H}_E)$ embedded in the full projective space.
- A near-degenerate manifold \mathcal{M}_C similarly defines a low-dimensional subspace whose projectivization $\mathbb{P}(\mathcal{M}_C)$ is where much of the “interesting” dynamics takes place when energies outside the manifold are energetically suppressed.

This geometric point of view becomes very powerful in:

- **Adiabatic evolution** with degeneracies:
 - In a non-degenerate case, slow adiabatic evolution can produce a Berry phase (scalar phase factor).
 - In a degenerate manifold, adiabatic evolution produces a **non-Abelian Berry connection** (Wilczek–Zee), i.e. a unitary transformation acting within the degenerate manifold.
- **Band theory**:
 - In condensed matter, a “band” of energies that is isolated from others defines a **vector bundle** over momentum space. Near-degenerate bands can merge and split, with topology encoded in Berry curvature, Chern numbers, etc.

For My few-qubit Hamiltonians: the “manifold” language is mostly a way to say “small subspace spanned by a cluster of eigenstates we care about.” But the geometric concepts (Berry phases, adiabatic transport, non-Abelian holonomies) become relevant if you imagine varying parameters in H continuously.

8. Degeneracies and level statistics (eigenvalue spacings)

You have explicitly mentioned:

- eigenvalue spacings,

- eigenvector amplitude patterns,
- and dominant basis states.

From the perspective of **eigenvalue spacings**:

- In random-matrix-like Hamiltonians with no special symmetry, exact degeneracies are rare.
- Instead, you see **level repulsion**: very small spacings are suppressed (Wigner–Dyson statistics).
- Clusters of very small spacings can indicate:
 - approximate symmetry,
 - some nearly conserved quantity,
 - or parameter regimes where certain couplings are weak.

In My analysis framework:

- You can scan the sorted spectrum $E_0 \leq E_1 \leq \dots$, compute Δ_j , and detect regions where successive Δ_j are below ε .
- Each such region defines a candidate near-degenerate manifold \mathcal{M}_C .
- Then you can inspect the **eigenvector amplitudes** in my preferred computational basis (Pauli-Z basis, etc.) to see whether these near-degenerate eigenstates share the same “dominant basis states.”
 - If multiple eigenstates all have large weight on the same few computational-basis configurations, that often indicates a physically meaningful manifold (e.g. “low-energy spin configurations,” “almost-localized states,” etc.)

9. Effective Hamiltonians on near-degenerate manifolds

Once you identify a near-degenerate manifold \mathcal{M}_C with projector P_C , you can systematically construct an **effective Hamiltonian** acting only on that manifold.

A standard approach is the **Schrieffer–Wolff transformation (SW)**:

- You choose an anti-Hermitian generator S such that you approximately block-diagonalize the full Hamiltonian:

$$\tilde{H} = e^S H e^{-S} \approx \begin{pmatrix} H_{\text{eff}} & 0 \\ 0 & H_{\text{rest}} \end{pmatrix},$$

where the upper-left block acts on \mathcal{M}_C .

- The effective Hamiltonian is then

$$H_{\text{eff}} \approx P_C \tilde{H} P_C,$$

computed perturbatively in the small parameter(s) controlling the coupling between \mathcal{M}_C and its complement.

Intuitively, SW “integrates out” high-energy states and yields an effective low-energy description in terms of the near-degenerate manifold.

For My application: if My algorithm identifies a near-degenerate manifold, you can:

- Focus numerically on that subspace.
- Construct an effective description (e.g. in terms of a smaller Pauli Hamiltonian restricted to \mathcal{M}_C).
- Use that for classification, simulation, or as input to quantum algorithms (e.g., variational methods on a reduced subspace).

10. Summary: what to keep conceptually

1. Degeneracy

- Multiple independent eigenstates share the same eigenvalue.
- Mathematically: $\dim \mathcal{H}_E > 1$.
- Physically: multiple stationary states with the same energy.
- Often linked to symmetry; basis inside a degenerate eigenspace is not unique.

2. Near-degeneracy

- Eigenvalues form tight clusters: level splittings \ll typical spectral gaps.
- Treated effectively as degenerate for many purposes (perturbation theory, low-energy dynamics).
- Defined in practice via a numerical tolerance ε .

3. Manifolds

- The subspace spanned by a degenerate or near-degenerate set of eigenstates.
- Equipped with a projector P .
- Dynamics restricted to this subspace can often be described by an effective Hamiltonian H_{eff} .

4. Why it matters for My analyses

- Eigenvalue spacings reveal clusters \rightarrow candidate near-degenerate manifolds.
- Inside these manifolds, eigenvector amplitude patterns and dominant basis states tell you what “effective degrees of freedom” the physics is using.
- Degenerate / quasi-degenerate structure explains where my Hamiltonian is “almost symmetric” or where different configurations are energetically indistinguishable.

5. Dominant Basis-State Structure

Eigenvectors often show low-entropy structure:

$$|\psi\rangle = \alpha_a |a\rangle + \alpha_b |b\rangle + (\text{small terms})$$

with:

$$|\alpha_a|^2 + |\alpha_b|^2 \gg 0.9$$

This is the fingerprint of "physics-native information units".

Description of Dominant Basis-State Structure

I will treat "dominant basis-state structure" in the same setting as before: a finite-dimensional Hilbert space (few-qubit system), with a fixed reference basis (typically the computational basis).

1. Setup: eigenvectors expressed in a fixed reference basis

Let

- H be a Hermitian Hamiltonian on an n -qubit Hilbert space $\mathcal{H} \cong (\mathbb{C}^2)^{\otimes n}$.
- $\{|z\rangle\}_{z \in \{0,1\}^n}$ be the **computational basis**, where $z = z_1 z_2 \dots z_n$ is a bitstring and $|z\rangle = |z_1\rangle \otimes \dots \otimes |z_n\rangle$.

An eigenvector $|\psi_j\rangle$ of H is then expanded as

$$|\psi_j\rangle = \sum_{z \in \{0,1\}^n} c_{j,z} |z\rangle,$$

with complex amplitudes $c_{j,z}$ satisfying normalization

$$\sum_z |c_{j,z}|^2 = 1.$$

The **basis-state structure** of $|\psi_j\rangle$ is exactly the pattern of these amplitudes $\{c_{j,z}\}$ across the basis states.

2. Dominant basis states: definition and intuition

Informally, **dominant basis states** for an eigenvector $|\psi_j\rangle$ are those computational basis states $|z\rangle$ that carry most of its probability weight:

- Define probabilities $p_{j,z} = |c_{j,z}|^2$.
- A basis state $|z\rangle$ is “dominant” if $p_{j,z}$ is large compared to a chosen reference scale (e.g. typical or average probability).

For an n -qubit system with 2^n basis states, the uniform “completely delocalized” vector would have

$$p_{j,z} \approx 2^{-n} \text{ for all } z.$$

In contrast, a highly **localized** eigenstate might have

$$p_{j,z_*} \approx 1, p_{j,z \neq z_*} \approx 0,$$

meaning that a single basis state $|z_*\rangle$ dominates almost entirely.

A practical definition (for algorithms like my v0.3 / v0.4) often looks like:

- Choose a threshold θ (for example $\theta = \alpha 2^{-n}$ with $\alpha \gg 1$, or simply a fixed number like 0.05).
- Define the set of dominant basis states for eigenvector $|\psi_j\rangle$ as

$$\mathcal{D}_j = \{z \mid p_{j,z} = |c_{j,z}|^2 \geq \theta\}.$$

This identifies which computational configurations $|z\rangle$ significantly contribute to $|\psi_j\rangle$.

Key idea: “dominant basis-state structure” means the **pattern** of which $|z\rangle$ ’s lie in \mathcal{D}_j , and how their weights are distributed and organized.

3. Localized vs delocalized eigenstates

Dominant basis-state structure gives a quantitative way to talk about **localization**:

1. Strongly localized eigenstates

- Only a few basis states have significant weight.
- $|\mathcal{D}_j|$ is small.
- Example: $|\psi_j\rangle \approx |z_*\rangle$ plus small corrections.

Physical interpretation:

- The eigenstate is close to a classical configuration $|z_*\rangle$ of spins/qubits.
- Measurements in the computational basis will almost always return the same outcome z_* .

2. Delocalized eigenstates

- Many basis states contribute with comparable weight.
- $|\mathcal{D}_j|$ is large, possibly of order 2^n .

- Example: random vector with approximately $p_{j,z} \approx 2^{-n}$.

Physical interpretation:

- No single classical configuration dominates.
- Measurement outcomes in the computational basis are broadly distributed.

A common quantitative measure is the **inverse participation ratio (IPR)**:

$$\text{IPR}_j = \sum_z |c_{j,z}|^4.$$

- $\text{IPR}_j = 1$ if the state is exactly one basis state.
- $\text{IPR}_j = 2^{-n}$ for a perfectly uniform state.
- Effective number of significantly contributing basis states is roughly $1/\text{IPR}_j$.

This “effective number” corresponds directly to $|\mathcal{D}_j|$ in a threshold-based picture.

4. Structure across multiple eigenstates: patterns and manifolds

When you look not just at a single eigenvector but at a set of them (e.g. a near-degenerate manifold), **patterns** in dominant basis states are very informative.

Let \mathcal{M} be a set of eigenstate indices (e.g. cluster of near-degenerate eigenvalues), and define for each $j \in \mathcal{M}$ the dominant set \mathcal{D}_j .

You can then consider:

- The union of dominant states:

$$\mathcal{D}_{\mathcal{M}} = \bigcup_{j \in \mathcal{M}} \mathcal{D}_j.$$

- Overlaps between dominant sets:

$$\mathcal{O}_{j,k} = \mathcal{D}_j \cap \mathcal{D}_k.$$

Interpretation:

1. Shared dominant basis states

- If multiple eigenstates share the same dominant $|z\rangle$'s (large overlaps $\mathcal{O}_{j,k}$), this suggests:
 - These eigenstates describe different quantum superpositions of the same underlying “classical configurations.”
 - They often form a natural **manifold** in which dynamics is largely confined.

2. Disjoint dominant patterns

- If \mathcal{D}_j and \mathcal{D}_k are almost disjoint for different eigenstates, those states describe entirely different regions of configuration space.
- This often correlates with different **phases**, **domains**, or **symmetry sectors** in more complex systems.

From the viewpoint of my algorithmic pipeline, “dominant basis-state structure” is a way to:

- Cluster eigenstates based on which classical configurations dominate their amplitude profiles.
- Identify effective low-dimensional subspaces that may be physically meaningful (e.g. low-energy classical configurations, metastable states, etc.).

5. Energy–configuration relationships

The dominant basis states tell you how the Hamiltonian’s **energy landscape** looks in the chosen basis:

- Consider the diagonal of H in the computational basis:

$$H_{zz} = \langle z | H | z \rangle.$$

- If off-diagonal couplings are not too large, eigenstates with strong peaks on certain $|z\rangle$ typically correspond to **local minima** (or other special points) in the energy landscape defined by H_{zz} .
- Then:
 - Dominant basis states identify “valleys” or “basins” in configuration space.
 - Near-degenerate eigenstates sharing dominant $|z\rangle$ ’s can represent different quantum superpositions localized around the same basin.

In a more strongly quantum regime (large off-diagonal couplings), the relationship is less classical but still informative:

- Highly delocalized eigenstates may correspond to regions where the quantum dynamics is “ergodic” over many configurations.
- Partially localized eigenstates (few dominant configurations) may indicate emergent integrals of motion, approximate symmetries, or constraints.

6. Symmetry, constraints, and dominant structure

Symmetries (e.g. parity, total spin, conserved excitation number) strongly constrain dominant basis-state structure:

1. Hard constraints

- If an operator Q commutes with H , and $|\psi_j\rangle$ is also an eigenstate of Q , then its support in the computational basis is restricted to basis states with a specific eigenvalue of Q .

- Example: if total excitation number \hat{N} is conserved and $|\psi_j\rangle$ has eigenvalue N , then $c_{j,z} = 0$ unless the bitstring z has Hamming weight N .
- Dominant basis states then all live inside the fixed-weight subspace.

2. Approximate symmetries

- If a symmetry is only approximate, the state may still be mostly supported on basis states satisfying that symmetry, with small weight elsewhere.
- Dominant basis-state structure reveals this approximate selection: non-dominant basis states often break the symmetry.

For a practical classifier:

- You can project basis states into “feature space” based on symmetry-related counts (Hamming weight, spin domain walls, etc.).
- Then inspect whether dominant basis states of different eigenstates occupy similar regions in that feature space.

7. Basis-state structure as a diagnostic for phases and regimes

In many-body physics, examining the dominant basis-state structure of eigenstates is a standard way to distinguish between different dynamical regimes:

1. Ergodic / chaotic regime

- Eigenstates resemble random vectors in the computational basis.
- No small set of dominant basis states; $|\mathcal{D}_j|$ is large.
- Participation ratio is large; entropy is high.

2. Localized regime (e.g. Anderson or many-body localization)

- Eigenstates are sharply peaked on a small set of basis states.
- Clear dominant basis-state structure, often concentrated around a single classical configuration.
- Participation ratio is low; entropy is low.

3. Symmetry-broken or ordered phases

- Low-energy eigenstates often have dominant basis states corresponding to different macroscopically distinct patterns (e.g. all spins up vs all spins down).
- Symmetric combinations (like $|\uparrow\uparrow \dots \uparrow\rangle \pm |\downarrow\downarrow \dots \downarrow\rangle$) have two dominant basis states.

My framework of “dominant basis-state structure” can be seen as a compact way of detecting which regime you are in, even for small or synthetic Hamiltonians.

8. Quantitative indicators beyond a simple threshold

While a simple amplitude threshold defines \mathcal{D}_j , you can refine the analysis with:

1. Sorted amplitude profiles

- Sort $|c_{j,z}|^2$ in descending order: $p_{j,1}^\downarrow \geq p_{j,2}^\downarrow \geq \dots$.
- Inspect how fast the cumulative sum

$$S_j(K) = \sum_{k=1}^K p_{j,k}^\downarrow$$

approaches 1.

- The minimal K such that $S_j(K) \geq \eta$ (e.g. $\eta = 0.9$) is the effective number of dominant basis states at confidence η .

2. Shannon entropy

- For eigenstate $|\psi_j\rangle$, define the Shannon entropy in the computational basis:

$$S_j = - \sum_z p_{j,z} \log p_{j,z}$$

- Low entropy \rightarrow strong dominance of a few basis states.
- High entropy \rightarrow delocalized, many comparable contributions.

3. Clustering of dominant bitstrings

- Take the dominant set \mathcal{D}_j , view bitstrings $z \in \mathcal{D}_j$ in Hamming space.
- Measure e.g. pairwise Hamming distances.
- Small typical distances indicate that dominant states form a “cluster” (e.g., only a few spin flips apart).
- Large distances indicate that dominant configurations are structurally distinct.

These diagnostics give a richer, more “deep” description of dominant basis-state structure than a single yes/no classification.

9. Relation to my pipeline: eigenvectors, manifolds, and classification

Connecting this back to the bullets you listed (eigenvalue spacings, eigenvector amplitude patterns, dominant basis states):

1. Eigenvalue spacings

- Identify near-degenerate clusters of eigenvalues (manifolds).

- These define candidate subspaces \mathcal{M}_C .

2. Eigenvector amplitude patterns

- For each eigenstate in a cluster, inspect the vector $\{c_{j,z}\}_z$.
- Look for structured patterns (localization, symmetry-related patterns, etc.).

3. Dominant basis states

- Extract \mathcal{D}_j for each eigenstate.
- Study overlaps $\mathcal{O}_{j,k}$, union \mathcal{D}_M , Hamming distances, etc.
- Use these structures as features to:
 - Characterize the manifold (e.g., “this manifold lives on these 5 bitstrings”).
 - Build effective models (reduced Hamiltonians on \mathcal{D}_M).
 - Feed into classical or quantum classifiers.

In short: **dominant basis-state structure** is the distilled information about “which classical configurations matter most” for each eigenstate or manifold. It is the bridge between:

- the abstract spectral structure of H , and
- concrete, interpretable classical labels or patterns in the computational basis.

6. Time Evolution (v0.5)

Time evolution is governed by:

$$U(t) = \exp(-i H t)$$

For a Pauli-sum Hamiltonian:

$$H = \sum_k h_k P_k$$

Description of Time Evolution (v0.5)

I will describe time evolution in exactly the same “spectral + Pauli + manifolds + dominant basis states” language as my earlier v0.3–v0.5 notes.

1. Schrödinger dynamics and the time-evolution operator

Given a time-independent Hamiltonian H , the state $|\phi(t)\rangle$ evolves according to the Schrödinger equation

$$i \frac{d}{dt} |\phi(t)\rangle = H |\phi(t)\rangle$$

(with $\hbar = 1$ in my conventions).

The formal solution is

$$|\phi(t)\rangle = U(t) |\phi(0)\rangle,$$

where the **time-evolution operator** is

$$U(t) = e^{-iHt}.$$

Everything in v0.5 about “dynamics” or “time evolution” is fundamentally the structure of this operator and its action on chosen initial states.

2. Using the spectral decomposition of H

From my earlier sections:

- The Hamiltonian has the spectral decomposition

$$H = \sum_j E_j |\psi_j\rangle\langle\psi_j|,$$

where $H |\psi_j\rangle = E_j |\psi_j\rangle$ and $\{\psi_j\}$ is an orthonormal eigenbasis.

Then the exponential is trivial in that basis:

$$U(t) = e^{-iHt} = \sum_j e^{-iE_j t} |\psi_j\rangle\langle\psi_j|.$$

So:

- Each eigenstate $|\psi_j\rangle$ is a **stationary state** up to a phase:

$$U(t) |\psi_j\rangle = e^{-iE_j t} |\psi_j\rangle.$$

- Time evolution only changes **phases** between different energy components.

Now expand an arbitrary initial state $|\phi(0)\rangle$ in the eigenbasis:

$$|\phi(0)\rangle = \sum_j a_j |\psi_j\rangle, a_j = \langle\psi_j|\phi(0)\rangle.$$

Time evolution is then

$$|\phi(t)\rangle = \sum_j a_j e^{-iE_j t} |\psi_j\rangle.$$

Key point: time evolution is entirely governed by the coefficients $\{a_j\}$ and the eigenvalues $\{E_j\}$. All dynamical features are encoded in:

- **Eigenvalues:** phases $e^{-iE_j t}$,
- **Eigenvectors:** how they overlap with the initial state and with the measurement basis.

This is exactly why my v0.3/v0.4 spectral analysis is a perfect stepping stone for v0.5.

3. Time scales from eigenvalue spacings

Define energy differences (transition frequencies)

$$\omega_{jk} = E_j - E_k.$$

In observables or amplitudes, phases appear as factors $e^{-i\omega_{jk} t}$. So:

- Large $|\omega_{jk}|$ \rightarrow **fast oscillations** (short time scale),
- Small $|\omega_{jk}|$ \rightarrow **slow beats** (long time scale).

In particular:

- If you have a **near-degenerate manifold** (cluster) of eigenvalues $E_j \approx E_k$, then the width of that cluster is small, and you get slow dynamics associated with coherence within that manifold.
- If there are large gaps between manifolds, those transitions contribute very fast, often effectively averaged-out oscillations.

So my **eigenvalue spacings** immediately define the **hierarchy of dynamical time scales**:

- Within a near-degenerate manifold: slow, structured evolution (beats).
- Between well-separated manifolds: fast oscillations, often washed out in coarse-grained measurements.

4. Time evolution in the computational basis

Let $\{|z\rangle\}$ be the computational basis, and expand eigenstates as:

$$|\psi_j\rangle = \sum_z c_{j,z} |z\rangle.$$

Initial state:

- Often you choose $|\phi(0)\rangle = |z_0\rangle$ (a computational basis state) or a simple superposition of a few such states.

Then

$$a_j = \langle \psi_j | \phi(0) \rangle = \sum_z c_{j,z}^* \langle z | \phi(0) \rangle.$$

If $|\phi(0)\rangle = |z_0\rangle$, then simply $a_j = c_{j,z_0}^*$.

Time-evolved state in the computational basis:

$$|\phi(t)\rangle = \sum_j a_j e^{-iE_j t} |\psi_j\rangle = \sum_z \left(\sum_j a_j e^{-iE_j t} c_{j,z} \right) |z\rangle.$$

So the amplitude on basis state $|z\rangle$ is

$$\phi_z(t) = \sum_j a_j c_{j,z} e^{-iE_j t}.$$

Measurement probabilities in the computational basis:

$$P(z, t) = |\phi_z(t)|^2.$$

Thus, time evolution is a **multi-frequency interference pattern** built from:

- Eigenvalues E_j ,
- Eigenvectors $c_{j,z}$,
- Initial overlaps a_j .

This is where my **dominant basis-state structure** and **near-degenerate manifolds** directly control visible dynamics.

5. Role of dominant basis-state structure in dynamics

If an eigenstate $|\psi_j\rangle$ is strongly localized on a small set of basis states (dominant basis states \mathcal{D}_j), then:

- That eigenstate contributes significantly only to those $|z\rangle \in \mathcal{D}_j$.
- If my initial state overlaps mostly with that eigenstate (or with a small group of such eigenstates), time evolution will be largely confined to those dominant basis states, up to interference.

In particular:

- If $|\phi(0)\rangle = |z_0\rangle$ and $|z_0\rangle$ is a dominant basis state for a subset of eigenstates in a near-degenerate manifold, then:
 - The coefficients a_j are large only for those eigenstates.
 - Dynamics will primarily involve slow phase rotations inside that manifold, modulating amplitudes across a small set of dominant basis states.

Conceptually:

- **Dominant basis states** tell you *where* in the configuration space the dynamics lives.
- **Eigenvalue spacings** (within and between manifolds) tell you *how fast* things move.

6. Degeneracies, near-degeneracies, and slow dynamics

For an exactly degenerate eigenvalue cluster E_α :

- Inside that eigenspace:

$$U(t)P_\alpha = e^{-iE_\alpha t}P_\alpha,$$

so any state within the degenerate subspace only picks up a global phase; **relative phases inside the degenerate space do not evolve** under H .

For a **near-degenerate manifold** with energies $E_j = E_\alpha + \delta_j$, where $|\delta_j|$ are small:

- In a frame rotating with $e^{-iE_\alpha t}$, you see slow evolution governed by the small splittings δ_j .
- Physically, my state executes slow beats within the manifold.

Thus, in v0.5, if you identify these near-degenerate clusters, you know:

- On short time scales: the manifold behaves almost as if it were degenerate (coherent “block” dynamics).
- On long time scales: small energy splittings gradually become visible as slow oscillations between the manifold’s eigenmodes.

7. Observables and time evolution

Given an observable O , its expectation in state $|\phi(t)\rangle$ is

$$\langle O \rangle(t) = \langle \phi(t) | O | \phi(t) \rangle.$$

In the eigenbasis of H :

$$\langle O \rangle(t) = \sum_{j,k} a_j^* a_k e^{i(E_j - E_k)t} \langle \psi_j | O | \psi_k \rangle.$$

- Diagonal terms ($j = k$): constant contributions.
- Off-diagonal terms ($j \neq k$): oscillatory contributions with frequencies $\omega_{jk} = E_j - E_k$.

Especially interesting:

1. Survival amplitude and probability

- Survival amplitude:

$$A_{\text{surv}}(t) = \langle \phi(0) | \phi(t) \rangle = \sum_j |a_j|^2 e^{-iE_j t}.$$

- Survival probability:

$$P_{\text{surv}}(t) = |A_{\text{surv}}(t)|^2.$$

2. This encodes how likely the system is to “return” to the initial state; its frequency content is directly tied to eigenvalue spacings.

3. Local Pauli observables

- For qubit k , you might track $\langle Z_k(t) \rangle$, $\langle X_k(t) \rangle$, etc.
- These are natural to measure on IBM hardware and give time-series that reflect interference between energy levels.

4. Time-averages

- Long-time average of $\langle O \rangle(t)$ often projects onto the **diagonal in the energy basis** (the so-called diagonal ensemble).
- This essentially keeps only the $|a_j|^2$ weights and drops off-diagonal coherences, which is closely related to my spectral weights.

In a classification context, **time traces of simple observables** are a very compact fingerprint of the full spectral structure.

8. Digital time evolution with Pauli Hamiltonians (IBM backend, v0.5)

My v0.5 prototype uses a Hamiltonian written as a sum of Pauli strings, e.g.

$$H = \sum_{\ell=1}^L \alpha_\ell P_\ell,$$

with P_ℓ being tensor products of I, X, Y, Z , as in my example:

- YZX, IIX, YII, IZZ, XYZ , etc.

The exact time-evolution operator is

$$U(t) = e^{-iHt} = e^{-it \sum_\ell \alpha_\ell P_\ell}.$$

8.1 Commuting vs non-commuting terms

- If all Pauli terms P_ℓ commute, then:

$$U(t) = \prod_\ell e^{-i\alpha_\ell P_\ell t}$$

exactly.

- If they do not commute (the generic case), we approximate using **Trotter–Suzuki decompositions**, e.g. first-order Trotter:

$$U(t) \approx \left(\prod_{\ell} e^{-i\alpha_{\ell} P_{\ell} \Delta t} \right)^m,$$

where $t = m\Delta t$. Higher-order formulas can reduce Trotter error.

8.2 Implementing $e^{-i\theta P}$ on hardware

Each gate of the form $e^{-i\theta P}$ (with P a Pauli string) is implemented via:

1. **Basis change:** Map each non-Z Pauli to Z using single-qubit rotations:
 - o $X \rightarrow Z$ via Hadamard,
 - o $Y \rightarrow Z$ via $S^{\dagger}H$, etc.
2. **Entangling CNOT chain:** For multi-qubit Z-strings, use CNOTs to concentrate parity information onto a single qubit.
3. **Single-qubit rotation:** Apply $R_z(2\theta)$ on the target qubit.
4. **Undo:** Reverse CNOTs and basis changes.

Thus, v0.5's "backend-ready" circuits are essentially stacks of these building blocks arranged according to the Trotterized product.

8.3 Time grid and measurements

In practice in v0.5:

- Choose a set of times $\{t_0, t_1, \dots, t_T\}$.
- For each t_{ℓ} :
 - o Construct a circuit approximating $U(t_{\ell})$.
 - o Prepare initial state $| \phi(0) \rangle$ (e.g. a computational basis state).
 - o Apply $U(t_{\ell})$.
 - o Measure in the computational basis (or in rotated bases for different observables).
- Collect statistics (bitstrings) from many shots.

From these measurement data you reconstruct:

- $P(z, t)$,
- expectation values of local Paulis,
- survival probabilities, etc.

All of that is the **empirical realization** of the theoretical time evolution described above.

9. Linking v0.5 dynamics back to spectral features (v0.3/v0.4)

Conceptually, v0.5 is the **dynamical layer** built on top of the static spectral analysis:

1. v0.3 / v0.4:
 - Compute eigenvalues E_j .
 - Compute eigenvectors $|\psi_j\rangle$.
 - Analyze eigenvalue spacings, near-degenerate manifolds, dominant basis states.
2. v0.5:
 - Use the same H (same Pauli coefficients) to build a dynamical simulator.
 - Observe how chosen initial states evolve in time:
 - How fast they dephase.
 - Which basis states become populated.
 - How observables oscillate or relax.

The deep link:

- The **frequencies** appearing in time series of observables are exactly the **differences of eigenvalues**.
- The **amplitudes** of those frequencies are set by the overlaps a_j and matrix elements $\langle\psi_j|O|\psi_k\rangle$, which are themselves determined by the eigenvector structure (including dominant basis states).
- Near-degenerate manifolds produce **slow components** in the dynamics (long-lived coherences).
- Strongly localized eigenstates produce **stable patterns** in basis-state probabilities over time.

So, time evolution in v0.5 is not an extra, unrelated feature; it is simply the **Fourier-space (time-domain) manifestation** of all the spectral structure you already extracted in v0.3/v0.4.

10. Conceptual summary

- **Time evolution** is governed by $U(t) = e^{-iHt}$.
- In the eigenbasis, each eigenstate just picks up a phase $e^{-iE_j t}$; dynamics is interference between these phases.
- **Eigenvalue spacings** determine the characteristic frequencies and time scales in observables.
- **Near-degenerate manifolds** generate slow dynamics and long-lived coherences.

- **Dominant basis-state structure** determines which computational basis states actually participate in the evolution, and how local or delocalized the dynamics is.
- On hardware (v0.5), time evolution is implemented by Trotterized exponentials of Pauli strings, measured over a time grid to reconstruct dynamical fingerprints of the Hamiltonian.

If you like, next step could be: take my concrete 3-qubit v0.5 Hamiltonian, pick a simple initial basis state (say $|1000\rangle$), and I can explicitly write out:

- its expansion in the eigenbasis,
- the exact analytic expression for $|\phi(t)\rangle$,
- and the corresponding probabilities $P(z,t)$ for a few key basis states, showing how eigenvalues and dominant basis states shape the dynamics.

7. Trotter Approximation (v0.5)

Since exact exponentiation is not feasible on hardware:

$$\exp(-iHt) \approx \prod_k \exp(-i h_k P_k \Delta t)$$

Trotter error:

$$\|U(t) - U_{\text{trotter}}(t)\| = O(t^2 / m)$$

My v0.5 experiment applies this both *locally* and *on IBM Heron*.

Description of Trotter Approximation (v0.5)

Here is a deep, v0.5-style description of the Trotter approximation, tailored to my setting: few-qubit Pauli Hamiltonians and IBM backend implementation.

1. Why we need Trotter approximation in v0.5

In v0.5, my Hamiltonian is written as a sum of Pauli strings:

$$H = \sum_{\ell=1}^L \alpha_\ell P_\ell,$$

where each P_ℓ is a tensor product of I, X, Y, Z on n qubits, e.g.

$$H = 0.3454 YZX + 0.3549 IIX - 0.6469 YII - 0.3957 IZZ + 0.4102 XYZ,$$

as in my SparsePauliOp example.

The exact time-evolution operator is

$$U(t) = e^{-iHt} = e^{-it \sum_{\ell=1}^L \alpha_\ell P_\ell}.$$

On actual hardware, we do *not* have a native gate that implements “exponentiate the entire sum” in one shot. What we *do* know how to implement efficiently are exponentials of *single* Pauli strings:

$$e^{-i\theta P}, P \in \{I, X, Y, Z\}^{\otimes n}.$$

So the problem is:

How to approximate $e^{-it \sum_{\ell=1}^L \alpha_\ell P_\ell}$ using only products of simpler gates $e^{-i\theta P_\ell}$?

The Trotter (and more generally Trotter–Suzuki) formulas are precisely the mathematical tools that allow us to do this systematically and quantify the approximation error.

2. The core idea: breaking a sum of non-commuting terms

Start with the simplest nontrivial case: two terms

$$H = A + B.$$

If A and B commute ($[A, B] = 0$), then

$$e^{-i(A+B)t} = e^{-iAt} e^{-iBt} = e^{-iBt} e^{-iAt},$$

exactly.

But in general, my Pauli strings do **not** commute, so

$$e^{-i(A+B)t} \neq e^{-iAt} e^{-iBt}.$$

The **Trotter product formula** gives an approximation:

- First-order (Lie–Trotter):

$$e^{-i(A+B)t} \approx (e^{-iA\Delta t} e^{-iB\Delta t})^m, \Delta t = t/m.$$

Intuition:

- Instead of trying to do “ $A + B$ ” in one go, we take many small steps Δt .
- In each small step, we first apply evolution under A for Δt , then under B for Δt .
- As $\Delta t \rightarrow 0$ and $m \rightarrow \infty$, this product converges to the true exponential.

This generalizes straightforwardly to many terms $\{H_\ell\}$.

3. General Trotter formula for many Pauli terms

Let

$$H = \sum_{\ell=1}^L H_\ell, H_\ell = \alpha_\ell P_\ell.$$

3.1 First-order (v0.5 basic version)

The **first-order Trotter formula** is:

$$U(t) = e^{-iHt} \approx \left(\prod_{\ell=1}^L e^{-iH_\ell \Delta t} \right)^m = \left(\prod_{\ell=1}^L e^{-i\alpha_\ell P_\ell \Delta t} \right)^m, \Delta t = t/m.$$

So the Trotter circuit for time t with m steps is:

1. For step $s = 1, \dots, m$:

- For each term $\ell = 1, \dots, L$:
 - Apply gate $e^{-i\alpha_\ell P_\ell \Delta t}$.

This yields a depth $\sim mL$ structure composed entirely of “Pauli rotation” blocks $e^{-i\theta P_\ell}$.

This is precisely what you would implement on the IBM backend in v0.5: each “Trotter step” is a fixed pattern of Pauli-rotation subcircuits, repeated m times.

3.2 When do we trust it?

The approximation becomes better as:

- Δt becomes smaller (larger m),
- the commutators between different H_ℓ become smaller in operator norm.

We now quantify this.

4. Where the error comes from: Baker–Campbell–Hausdorff (BCH)

The BCH formula tells us, schematically, that for small Δt :

$$e^{-iA\Delta t} e^{-iB\Delta t} = \exp \left(-i(A + B)\Delta t - \frac{(\Delta t)^2}{2} [A, B] + \mathcal{O}(\Delta t^3) \right).$$

So a single Trotter step for $H = A + B$ has an effective Hamiltonian:

$$H_{\text{eff}} = A + B + \mathcal{O}(\Delta t),$$

and the **error per step** is of order $\mathcal{O}(\Delta t^2)$.

Over m steps:

- Total time $t = m\Delta t$.
- Error accumulates (roughly) linearly in m , so total error scales like

$$\mathcal{O}(m\Delta t^2) = \mathcal{O}(t\Delta t) = \mathcal{O}(t^2/m).$$

For many terms $\{H_\ell\}$, the commutators become more involved, but the scaling is similar:

- First-order Trotter error:

$$\| U_{\text{Trotter}}(t) - e^{-iHt} \| \sim \mathcal{O} \left(\frac{t^2}{m} \sum_{\ell < k} \| [H_\ell, H_k] \| \right).$$

In Pauli language:

- Each commutator $[H_\ell, H_k] = \alpha_\ell \alpha_k [P_\ell, P_k]$.
- If P_ℓ and P_k commute, the commutator vanishes and they do **not** contribute to Trotter error.
- Error is entirely driven by *non-commuting* Pauli strings.

This is why grouping commuting terms and reordering terms can matter a lot.

5. Higher-order Trotter–Suzuki formulas

First-order Trotter is simple but only gives $\mathcal{O}(t^2/m)$ scaling. To reduce error without increasing gate count too dramatically, you can use higher-order formulas.

5.1 Second-order (symmetric) Trotter

For $H = \sum_\ell H_\ell$, the **second-order symmetric Trotter** step is:

$$U^{(2)}(\Delta t) = \left(\prod_{\ell=1}^L e^{-iH_\ell \Delta t/2} \right) \left(\prod_{\ell=L}^1 e^{-iH_\ell \Delta t/2} \right).$$

Over total time t :

$$U(t) \approx [U^{(2)}(\Delta t)]^m, \Delta t = t/m.$$

This cancels the leading-order error term and yields:

- Global error scaling $\sim \mathcal{O}(t^3/m^2)$.

Trade-off:

- A single second-order step has roughly twice as many exponentials as a first-order step.
- But for a given target error, you may need far fewer steps m , so total gate count can still be favorable.

5.2 Suzuki higher orders

More generally, **Suzuki formulas** recursively construct higher-order decompositions (4th, 6th, ...) with rapidly improving error scaling:

- k -th order: error $\sim \mathcal{O}(t^{k+1}/m^k)$.

In v0.5 you might not need to go beyond second order (hardware noise often dominates anyway), but conceptually:

- Higher order \rightarrow fewer Trotter steps for same accuracy,
- But each step is more complicated (longer sequence of exponentials).

6. Choosing Trotter step size and number of steps in practice

For v0.5, you conceptually choose m (number of Trotter steps) as a compromise between:

- **Trotter error** (unitary approximation error),
- **Hardware noise** (gate errors, decoherence),
- **Circuit depth and shot budget.**

Heuristics:

1. Fix a maximum simulation time t_{\max} you care about (e.g. due to decoherence limits).
2. Choose a target accuracy ϵ for the unitary approximation.
3. Use rough error scaling to choose m :
 - First-order: $m \sim \mathcal{O}\left(\frac{t_{\max} \sum \| [H_\ell, H_k] \|}{\epsilon}\right)$.
 - Second-order: $m \sim \mathcal{O}\left(\sqrt{\frac{t_{\max} \sum \| [[H_\ell, H_k], H_r] \|}{\epsilon}}\right)$ (schematically).
4. Translate m into total gate count and depth, check against backend constraints.

In small few-qubit demos (like my 3-qubit v0.5 prototype), you can simply try several m values (e.g. 1,2,4,8) and:

- Compare with classical exact simulation (to check Trotter error),
- Inspect the stability of key observables as m increases.

7. Circuits for $e^{-i\theta P_\ell}$: the hardware view

Trotter approximation would be useless if we could not efficiently implement $e^{-i\theta P_\ell}$.

For a single Pauli string P_ℓ , say $P_\ell = YZX$ on 3 qubits, the standard construction is:

1. **Local basis changes**
 - Map each non-Z Pauli to Z using 1-qubit gates:
 - $X \rightarrow Z$: apply Hadamard H .
 - $Y \rightarrow Z$: apply $S^\dagger H$.
 - After this, the string is converted to a Z-string, e.g. ZZZ on certain qubits.
2. **Entangling chain**
 - Use CNOTs to “collect” the parity of the Z-string onto a single qubit.
 - For a string on qubits q_1, q_2, \dots, q_r , you typically apply a chain of CNOTs.
3. **Single Z-rotation**

- Apply $R_z(2\theta)$ on the target qubit.

4. Undo

- Reverse the CNOT chain.
- Undo the basis changes.

This yields:

$$\text{Circuit}(P_\ell, \theta) \approx e^{-i\theta P_\ell},$$

up to global phases.

Trotterization in v0.5 essentially means: **stacking these subcircuits for different P_ℓ and for different steps.**

8. Trotter approximation and spectral structure

From the spectral perspective:

- Exact evolution:

$$U(t) = \sum_j e^{-iE_j t} |\psi_j\rangle\langle\psi_j|.$$

- Trotterized evolution:

$$U_{\text{Trotter}}(t) = \left(\prod_\ell e^{-i\alpha_\ell P_\ell \Delta t} \right)^m.$$

You can think of $U_{\text{Trotter}}(t)$ as exactly equal to the time evolution of a *slightly different Hamiltonian* H_{eff} :

$$U_{\text{Trotter}}(t) = e^{-iH_{\text{eff}} t},$$

where

$$H_{\text{eff}} = H + \delta H, \|\delta H\| \sim \mathcal{O}(t/m) \text{ (first order).}$$

So from the energy spectrum standpoint:

- Eigenvalues and eigenvectors of H_{eff} are small perturbations of those of H if δH is small.
- This means the Trotterized dynamics is effectively governed by a Hamiltonian whose spectrum $\{E'_j\}$ is close to $\{E_j\}$.

Consequences:

- **Near-degenerate manifolds** of H remain near-degenerate manifolds of H_{eff} , up to small splittings.

- **Dominant basis-state structures** of eigenstates are also perturbed only slightly (as long as δH is small compared to relevant gaps).
- Observed time series of observables will be close to what you'd get from the exact H , with small shifts in frequencies and amplitudes.

In other words: Trotterization slightly distorts the spectral features you analyzed in v0.3/v0.4, but in a controlled way, if you choose m reasonably.

9. How Trotter interacts with near-degenerate manifolds & dynamics

Linking back to my deeper structures:

1. Within a near-degenerate manifold

- Exact evolution depends sensitively on small differences $E_j - E_k$.
- Trotter error modifies these differences by $\delta(E_j - E_k)$ of order $\|\delta H\|$.
- If the manifold splittings are *extremely* small, Trotter error can be comparable to or larger than the physical splitting, and then it can noticeably distort slow dynamics (long-time beat patterns).
- For moderate near-degeneracy, a decent Trotter step size should preserve the qualitative structure (same manifold, similar beat frequencies).

2. Between well-separated manifolds

- Large gaps dominate; Trotter errors of moderate size do not fundamentally change “which manifold” you are in.
- Qualitative behavior (e.g. confinement of dynamics to a small manifold of dominant basis states) is robust.

So for v0.5:

- If you want to probe fine structure of very small splittings, you need higher *mor* higher-order formulas.
- If you only need coarse-grained dynamics and classification fingerprints, a modest Trotterization level is usually sufficient.

10. Compact conceptual summary (for my v0.5 documentation)

You can summarize Trotter approximation in my v0.5 notes roughly like this:

- The v0.5 Hamiltonian is a weighted sum of Pauli strings:

$$H = \sum_{\ell} \alpha_{\ell} P_{\ell}.$$

- The exact time-evolution operator $U(t) = e^{-iHt}$ cannot be implemented directly on IBM hardware.
- Trotter approximation replaces the single exponential by a product of exponentials of individual Pauli terms:

- First-order:

$$U(t) \approx \left(\prod_{\ell} e^{-i\alpha_{\ell} P_{\ell} \Delta t} \right)^m, \Delta t = t/m.$$

- Second-order (symmetric):

$$U(t) \approx \left(\prod_{\ell} e^{-i\alpha_{\ell} P_{\ell} \Delta t/2} \prod'_{\ell} e^{-i\alpha_{\ell} P_{\ell} \Delta t/2} \right)^m.$$

- Each gate $e^{-i\theta P_{\ell}}$ is decomposed into a sequence of single-qubit rotations and CNOTs.
- Trotter error is controlled by:
 - Time step Δt ,
 - Number of steps m ,
 - Commutators between Pauli terms.
- In the limit $m \rightarrow \infty$, the Trotter circuit converges to exact $U(t)$; for finite m , it simulates the dynamics of a slightly perturbed Hamiltonian $H_{\text{eff}} = H + \delta H$.
- As long as δH is small compared to relevant energy scales:
 - Eigenvalues and eigenvectors (including near-degenerate manifolds and dominant basis states) are only slightly deformed.
 - Time evolution fingerprints (oscillation frequencies, decay patterns, population transfers between dominant basis states) remain faithful to the original Hamiltonian.

8. Subspace Leakage (v0.5)

Given the v0.4 subspace:

$$S = \text{span}\{ |\phi_1\rangle, |\phi_2\rangle \}$$

Leakage is defined as:

$$L(t) = 1 - (|\langle\phi_1| \psi(t)\rangle|^2 + |\langle\phi_2| \psi(t)\rangle|^2)$$

with:

$$|\psi(t)\rangle = U(t) |\psi(0)\rangle$$

Low leakage = stable physics-native subspace.

Description of Subspace Leakage (v0.5)

I will describe “subspace leakage” in the same language as my other v0.5 sections: manifolds, projectors, time evolution, Trotter, and dominant basis states.

1. Setup: target manifold vs. rest of Hilbert space

Assume you have identified a **target subspace** (manifold) from my spectral analysis (v0.3/v0.4), for example:

- A low-energy near-degenerate manifold,
- A cluster of eigenstates with shared dominant basis states,
- Or any subspace you want to treat as “effective degrees of freedom”.

Formally:

- Let \mathcal{M} be the target subspace of the full Hilbert space \mathcal{H} .
- Let P be the projector onto \mathcal{M} :

$$P = \sum_{j \in \mathcal{M}} |\psi_j\rangle\langle\psi_j|.$$

- Let $Q = I - P$ be the projector onto the complement \mathcal{M}^\perp .

Then any state can be decomposed as:

$$|\phi\rangle = P|\phi\rangle + Q|\phi\rangle, \text{with } P|\phi\rangle \in \mathcal{M}, Q|\phi\rangle \in \mathcal{M}^\perp.$$

In the v0.5 context, you typically:

- Start with an initial state designed to live in \mathcal{M} , e.g. a superposition of eigenvectors in a chosen cluster, or a basis state whose spectral weight is dominated by that manifold.

- Evolve under some Hamiltonian (or its Trotterized approximation) and want to know: **how much of the state remains in \mathcal{M} and how much “leaks out” into \mathcal{M}^\perp ?**

This is precisely what subspace leakage measures.

2. Definition: leakage probability and fidelity to the manifold

Given a time-evolved state $|\phi(t)\rangle$, define:

- **In-subspace component:** $P |\phi(t)\rangle$,
- **Leakage component:** $Q |\phi(t)\rangle$.

The **survival probability in the subspace** is:

$$p_{\text{in}}(t) = \langle \phi(t) | P | \phi(t) \rangle = \| P | \phi(t) \rangle \|^2.$$

The **leakage probability** is then:

$$p_{\text{leak}}(t) = 1 - p_{\text{in}}(t) = \langle \phi(t) | Q | \phi(t) \rangle = \| Q | \phi(t) \rangle \|^2.$$

Interpretation:

- $p_{\text{leak}}(t) = 0$: the state is entirely within the target subspace \mathcal{M} .
- $p_{\text{leak}}(t) > 0$: part of the state has moved (leaked) into the complement \mathcal{M}^\perp .

In v0.5 terms:

- There is an **ideal** picture where dynamics stays inside the near-degenerate manifold (or some designed subspace).
- Subspace leakage quantifies how far the **actual** dynamics (including couplings, Trotter error, noise) deviates from that ideal.

3. Exact Hamiltonian: when is there no leakage?

Write the Hamiltonian in block form with respect to the P/Q decomposition:

$$H = \begin{pmatrix} H_{PP} & H_{PQ} \\ H_{QP} & H_{QQ} \end{pmatrix},$$

where:

- $H_{PP} = P H P$ acts within \mathcal{M} ,
- $H_{QQ} = Q H Q$ acts within \mathcal{M}^\perp ,
- $H_{PQ} = P H Q, H_{QP} = Q H P$ couple the two subspaces.

If:

1. The initial state lies completely inside \mathcal{M} :

$$|\phi(0)\rangle = P |\phi(0)\rangle,$$

2. And the Hamiltonian is **block-diagonal** in this decomposition:

$$H_{PQ} = PHQ = 0 \text{ and } H_{QP} = 0,$$

then it is straightforward to show:

- The subspace is **invariant** under time evolution:

$$|\phi(t)\rangle = e^{-iHt} |\phi(0)\rangle \in \mathcal{M} \forall t.$$

- There is **no subspace leakage**: $p_{\text{leak}}(t) = 0$ for all times.

So in the exact theory:

- **Subspace invariance \Leftrightarrow no off-diagonal blocks** H_{PQ}, H_{QP} .
- Any non-zero coupling H_{PQ} or H_{QP} can cause leakage.

In many realistic scenarios, H_{PQ} is not zero but **small**, so leakage is controlled but non-zero.

4. Leakage from small couplings: perturbative picture

Assume:

- Initial state $|\phi(0)\rangle$ lies entirely in \mathcal{M} .
- Couplings H_{PQ} and H_{QP} are “small” compared to energy gaps between \mathcal{M} and \mathcal{M}^\perp .

Then you can use time-dependent perturbation theory or a simple short-time expansion to estimate leakage.

For short times:

$$|\phi(t)\rangle = \left(1 - iHt - \frac{1}{2}H^2t^2 + \dots \right) |\phi(0)\rangle.$$

The leakage component to first non-trivial order is:

$$Q |\phi(t)\rangle \approx -itQHP |\phi(0)\rangle = -itH_{QP} |\phi(0)\rangle.$$

So the leakage probability behaves as:

$$p_{\text{leak}}(t) = \| Q |\phi(t)\rangle \|^2 \approx t^2 \| H_{QP} |\phi(0)\rangle \|^2 + \mathcal{O}(t^3).$$

Qualitatively:

- For very short times, leakage grows like t^2 with a prefactor controlled by the **norm of the coupling block** H_{QP} .
- If the energy gap Δ between \mathcal{M} and \mathcal{M}^\perp is large, transitions are further suppressed (Fermi's golden rule style).

So, **small inter-subspace couplings** and/or **large spectral gaps** keep subspace leakage under control.

5. Subspace leakage in v0.5 with Trotterized evolution

In v0.5 you do not apply the exact propagator e^{-iHt} , but a Trotterized approximation:

$$U_{\text{Trotter}}(t; m) \approx e^{-iHt},$$

built from products of exponentials of individual Pauli terms.

As discussed in my Trotter section:

- Trotterization can be thought of as simulating a slightly modified Hamiltonian:

$$U_{\text{Trotter}}(t; m) = e^{-i(H+\delta H)t},$$

where $\|\delta H\|$ scales like $\mathcal{O}(t/m)$ (first order) or better for higher orders.

Even if the **ideal Hamiltonian** H is block-diagonal (no exact leakage), the **effective Hamiltonian** $H + \delta H$ generally is not:

- δH will typically have **non-zero off-block components** $\delta H_{PQ}, \delta H_{QP}$.
- This induces **Trotter-induced subspace leakage**.

In total you have two contributions to leakage:

1. **Physical/inter-subspace leakage from** H_{PQ}, H_{QP} .
2. **Algorithmic leakage from Trotter error** via $\delta H_{PQ}, \delta H_{QP}$.

For a given number of Trotter steps m :

- If H itself has small but non-zero couplings H_{PQ} , Trotterization perturbs them slightly.
- If H is exactly block-diagonal (no leakage in principle), then any observed leakage in simulation / hardware comes purely from:
 - Trotter error,
 - State-preparation error,
 - Or hardware noise (see below).

So in v0.5, **subspace leakage is a sensitive diagnostic** of both physical couplings and numerical/implementation errors.

6. Hardware-level leakage vs. algorithmic subspace leakage

There are two conceptually different notions of “leakage” that are often conflated:

1. **Algorithmic subspace leakage** (what we mainly discuss here)
 - Defined relative to a chosen subspace \mathcal{M} inside the **model Hilbert space** (e.g. low-energy manifold, symmetry sector, etc.).
 - Happens even in ideal mathematics if $H_{PQ} \neq 0$.
 - Enhanced or created by Trotter error and imperfect control.
2. **Physical/hardware leakage**
 - In real qubits (e.g. transmons), there are higher energy levels $| 2\rangle, | 3\rangle, \dots$
 - The “qubit” logical space is the **computational subspace** spanned by $| 0\rangle, | 1\rangle$.
 - Imperfect pulses can drive population into those non-computational levels.
 - That is hardware leakage in a literal sense: leaving the two-level subspace.

In my v0.5 conceptual framework, you are primarily dealing with **algorithmic leakage between manifolds or subspaces of the model Hilbert space**. But in a full IBM backend implementation, both effects can be present and add up.

7. Subspace leakage viewed through dominant basis-state structure

My “dominant basis-state structure” gives a useful, concrete way to see subspace leakage in the computational basis.

Suppose:

- The target manifold \mathcal{M} is associated with a set of eigenstates whose dominant basis states belong to a certain region of configuration space (e.g. certain bitstrings, certain Hamming weights).
- At $t = 0$, my initial state is localized on those dominant basis states.

As time evolves:

- Ideal no-leakage dynamics: population remains, up to interference, within basis states that are dominantly supported by eigenstates in \mathcal{M} .
- Leakage: increasing probability mass on basis states that are **not** strongly represented in \mathcal{M} but belong dominantly to eigenstates outside \mathcal{M} .

So from a basis-state perspective:

- You can define a **basis-level approximation to P** by specifying a set of basis states $\mathcal{B}_{\mathcal{M}}$ that are dominant for the manifold.
- Then track:

$$p_{\text{in, basis}}(t) = \sum_{z \in \mathcal{B}_{\mathcal{M}}} P(z, t), p_{\text{leak, basis}}(t) = 1 - p_{\text{in, basis}}(t),$$

where $P(z, t)$ are computational-basis measurement probabilities from v0.5.

- This is a **coarse but measurement-friendly proxy** for the exact $p_{\text{leak}}(t)$ defined via projectors on eigenstates.

In small systems where you know the full spectrum and eigenvectors, you can compute the exact projector P and use it as a gold standard; on hardware you approximate it via dominant basis states.

8. Diagnosing and quantifying leakage in v0.5

In a purely classical simulation:

1. Choose a target manifold \mathcal{M} (e.g. a near-degenerate cluster).
2. Construct the projector $P = \sum_{j \in \mathcal{M}} |\psi_j\rangle\langle\psi_j|$.
3. Evolve initial state using:
 - Exact e^{-iHt} ,
 - Trotterized $U_{\text{Trotter}}(t; m)$.
4. Compute:

$$p_{\text{leak}}^{\text{exact}}(t), p_{\text{leak}}^{\text{Trotter}}(t; m).$$

5. Compare the two as functions of time and m .

This directly separates:

- **Intrinsic leakage** due to H_{PQ} ,
- **Trotter-induced leakage** due to δH .

On hardware (IBM backend):

- You do not have direct access to eigenvectors, but you can still:
 1. Use classical pre-analysis to identify a set of dominant basis states $\mathcal{B}_{\mathcal{M}}$ representing the manifold.
 2. Prepare an initial state localized in those states.
 3. Run Trotterized circuits for different times t , measure $P(z, t)$.

4. Estimate $p_{\text{in, basis}}(t)$ as above.
5. Increase Trotter steps m and see if $p_{\text{in, basis}}(t)$ stabilizes \rightarrow this checks Trotter convergence and reveals leakage trends.

9. Strategies to minimize subspace leakage

Conceptually, you can control subspace leakage in several ways.

9.1 Hamiltonian-level design

- Engineer H so that:
 - The target subspace is **approximately invariant** (small H_{PQ}).
 - There is a **large spectral gap** between \mathcal{M} and \mathcal{M}^\perp .
- Use **penalty terms** that energetically penalize states outside \mathcal{M} :

$$H \rightarrow H + \lambda Q,$$

with $\lambda > 0$ large, to suppress transitions.

9.2 Trotterization choices

- Use **more Trotter steps** or a **higher-order** Trotter–Suzuki formula to reduce $\| \delta H \|$.
- **Group commuting terms** in the Hamiltonian to reduce non-commuting cross-terms and thus Trotter error.
- Tune the total simulation time t so that:
 - You probe the dynamics of interest,
 - But do not accumulate excessive Trotter error that overwhelms the physical structure.

9.3 State preparation

- Start with initial states that are as close as possible to **eigenstates** or well-confined superpositions within \mathcal{M} .
- If initial states already have a large component in \mathcal{M}^\perp , you effectively start with non-zero leakage.

10. Conceptual summary (v0.5 wording)

You can phrase “Subspace Leakage (v0.5)” approximately like this in my documentation:

- We identify a **target manifold** \mathcal{M} of interest (e.g., a near-degenerate cluster of eigenstates with specific dominant basis-state patterns).
- Let P be the projector onto \mathcal{M} and $Q = I - P$ the projector onto its complement.
- For a time-evolved state $|\phi(t)\rangle$, we define the **leakage probability**:

$$p_{\text{leak}}(t) = \langle \phi(t) | Q | \phi(t) \rangle,$$

which measures how much population has left the target manifold.

- If the Hamiltonian is block-diagonal with respect to (P, Q) , and the initial state lies in \mathcal{M} , then $p_{\text{leak}}(t) = 0$ for all t .
- In general, off-diagonal blocks H_{PQ}, H_{QP} produce intrinsic leakage, scaling roughly as $t^2 \|H_{QP}\|^2$ at short times, regularized by spectral gaps.
- Trotter approximation replaces H by an effective Hamiltonian $H + \delta H$; its off-block components δH_{PQ} generate additional **Trotter-induced leakage**, even when H is ideally block-diagonal.
- In practice, we track subspace leakage by:
 - Computing $p_{\text{leak}}(t)$ via projectors in classical simulation, and/or
 - Using dominant basis-state structure to approximate P and monitor the population in the corresponding basis-state region on real hardware.
- Minimizing leakage involves:
 - Designing H so that the manifold is approximately invariant and gapped,
 - Choosing Trotter parameters (step number, order, grouping) to keep $\|\delta H\|$ small,
 - Preparing initial states well confined to \mathcal{M} .

9. Backend vs Local Agreement

Local simulation → exact amplitudes.

IBM backend → measurement samples with noise.

Structural agreement criterion:

$$\operatorname{argmax}_x P_{\text{local}}(x) = \operatorname{argmax}_x P_{\text{backend}}(x)$$

This held in v0.5 → strong evidence the structure is *physically real* and not a numerical artifact.

Description of Backend vs Local Agreement

Here I will treat **Backend vs Local Agreement** as a v0.5-level concept: comparing what my *hardware backend* (IBM device) does with what my *local model* (classical simulator + Pauli Hamiltonian + Trotter, etc.) predicts.

1. What we are comparing

1.1 “Local” model

My **local** (classical) model is everything you do on your own machine:

- A specified Hamiltonian

$$H = \sum_{\ell} \alpha_{\ell} P_{\ell},$$

where the α_{ℓ} and Pauli strings P_{ℓ} come from v0.3/v0.4 spectral analysis.

- Exact diagonalization:
 - Eigenvalues E_j ,
 - Eigenvectors $|\psi_j\rangle$,
 - Near-degenerate manifolds,
 - Dominant basis states.
- Exact (or very accurate) time evolution:

$$U_{\text{exact}}(t) = e^{-iHt},$$

and/or high-precision Trotter simulation using a noiseless simulator (e.g. Aer).

- Predicted measurement statistics:

$$P_{\text{local}}(z, t) = \Pr(\text{bitstring } z \text{ at time } t),$$

as well as time series of observables $\langle O(t) \rangle_{\text{local}}$.

This is my **theoretical reference**.

1.2 Backend

The **backend** is a specific IBM quantum device:

- You compile my circuits (Trotter steps, initial state prep, measurement) and send them to the backend.
- The device executes noisy gates, subject to:
 - Coherence times,
 - Gate errors,
 - Readout errors,
 - Calibration drift, etc.
- It returns empirical distributions

$$P_{\text{backend}}(z, t),$$

from finite shots, plus backend-level uncertainties.

Backend vs Local Agreement is exactly the degree to which

$$P_{\text{backend}}(z, t) \approx P_{\text{local}}(z, t)$$

and similarly for derived quantities like expectation values, survival probabilities, leakage metrics, etc.

2. Levels of agreement: what can “agree” and how

There are several layers at which you can talk about agreement.

2.1 Distribution-level agreement

At each time t , you have a full distribution over bitstrings:

- Local: $P_{\text{local}}(z, t)$.
- Backend: $P_{\text{backend}}(z, t)$.

You can compare them using, e.g.:

- Total variation distance:

$$D_{\text{TV}}(t) = \frac{1}{2} \sum_z |P_{\text{local}}(z, t) - P_{\text{backend}}(z, t)|.$$

- Kullback–Leibler divergence (asymmetric, but informative):

$$D_{\text{KL}}(P_{\text{backend}} \parallel P_{\text{local}}).$$

Small distances → strong distribution-level agreement.

2.2 Observable-level agreement

Rather than the full distribution, you often focus on **few-qubit observables**:

- Local:

$$\langle O(t) \rangle_{\text{local}} = \sum_z P_{\text{local}}(z, t) o(z),$$

where $o(z)$ is the eigenvalue of O on basis state $|z\rangle$.

- Backend:

$$\langle O(t) \rangle_{\text{backend}} = \sum_z P_{\text{backend}}(z, t) o(z).$$

Typical choices:

- Single-qubit Z_k, X_k (after basis change),
- Correlators $Z_i Z_j$,
- Survival probability, leakage measures, etc.

Agreement here means

$$\langle O(t) \rangle_{\text{backend}} \approx \langle O(t) \rangle_{\text{local}}$$

within error bars (shot noise + backend noise).

2.3 Dynamical / spectral fingerprints

Since v0.5 is explicitly about **time evolution**, a deeper level of agreement is:

- Take a time series for an observable O : $\{\langle O(t_0) \rangle, \dots, \langle O(t_T) \rangle\}$.
- Compare the **shapes**:
 - Oscillation frequencies,
 - Decay rates,
 - Beat patterns.

These time-series behaviours are directly tied to:

- Eigenvalue differences $\omega_{jk} = E_j - E_k$,
- Overlaps $a_j = \langle \psi_j | \phi(0) \rangle$,
- Matrix elements $\langle \psi_j | O | \psi_k \rangle$.

If backend and local data show the **same set of dominant frequencies** and **similar envelopes**, you have strong dynamical agreement, even if gate noise slightly damps amplitudes.

3. Sources of disagreement

Backend vs Local Agreement is never perfect. The gaps usually come from:

3.1 Model mismatch

My local Hamiltonian H is an **idealized model**:

- Pauli weights α_ℓ may not match the effective physical Hamiltonian realized by my circuit exactly.
- Transpilation and compilation optimizations may introduce gate ordering or decomposition differences not captured in my “paper Hamiltonian”.
- Crosstalk, residual ZZ-couplings, and other analog effects are not included in H .

Result: The effective Hamiltonian on the backend is $H_{\text{real}} \neq H$.

3.2 Trotter approximation

Even on the simulator, you approximate e^{-iHt} with a Trotterized product:

$$U_{\text{Trotter}}(t; m) = \left(\prod_{\ell} e^{-i\alpha_\ell P_\ell \Delta t} \right)^m, \Delta t = t/m.$$

This is exact only for $m \rightarrow \infty$.

- For finite m , the simulation corresponds to an **effective Hamiltonian**

$$H_{\text{eff}} = H + \delta H,$$

with $\| \delta H \|$ scaling like t/m (order-1) or better (higher order).

- If you use the **exact** model e^{-iHt} as my “local reference”, then Trotter error alone can create backend-local mismatch.

To be consistent, my **local reference** for backend comparison should be the same Trotter sequence you actually implement as gates.

3.3 Hardware noise

Physical backend adds:

- Depolarizing and dephasing noise,
- Relaxation (T1) and dephasing (T2),
- Gate calibration imperfections,
- Readout errors and bias,
- Drift over time.

These manifest as:

- Damping of oscillation amplitudes,
- Biases in measured probabilities,
- Additional effective terms beyond my target Hamiltonian.

3.4 Finite sampling (shot noise)

Even if backend implemented the ideal circuit, with a finite number of shots N , the observed frequencies $\hat{P}_{\text{backend}}(z, t)$ fluctuate around the true backend distribution.

- Statistical uncertainty scales as $\mathcal{O}(1/\sqrt{N})$.
- Agreement must be judged relative to these statistical error bars.

4. Agreement as a validation loop

In v0.5, “Backend vs Local Agreement” is not just a passive observation; it is a **validation loop**:

Define the model

- Choose H (Pauli coefficients) from my previous analysis (v0.3/v0.4).
- Decide on Trotter order and number of steps m .
- Fix initial state $| \phi(0) \rangle$ and a set of observables or basis-state probabilities you care about.

Local prediction

- Simulate exactly or via the same Trotter sequence:

$$| \phi_{\text{local}}(t) \rangle = U_{\text{Trotter}}(t; m) | \phi(0) \rangle.$$

- Compute:
 - $P_{\text{local}}(z, t)$,
 - $\langle O(t) \rangle_{\text{local}}$,
 - Survival, leakage, etc.

Backend experiment

- Compile the Trotter circuits for the same times $\{t_k\}$.
- Execute on backend, gather bitstrings, form empirical distributions $P_{\text{backend}}(z, t_k)$.
- Compute backend observables and statistics.

Compare

- For each time t_k , compute:
 - Distance between distributions,
 - Difference in $\langle O(t_k) \rangle$,

- Difference in survival / leakage, etc.
- Plot local vs backend curves over time.

Interpret discrepancies

- If differences are within expected noise margins:
 - Model is validated at that scale.
- If systematic deviations appear:
 - Revisit H (maybe important terms missing),
 - Increase Trotter steps or order,
 - Re-optimize calibration / error mitigation,
 - Or accept that hardware noise dominates beyond some t .

This is exactly where “Backend vs Local Agreement” becomes a **tool**, not just a statement: it tells you how far you can trust my hardware experiment as an implementation of my model Hamiltonian.

5. Role of manifolds and dominant basis states in this comparison

My v0.3/v0.4 machinery (manifolds, dominant basis states, leakage) gives **structured indicators** for agreement.

5.1 Manifold-level indicators

Suppose you have identified a near-degenerate manifold \mathcal{M} (projector P).

Locally (classical):

- Track subspace survival probability

$$p_{\text{in}}^{\text{local}}(t) = \langle \phi_{\text{local}}(t) | P | \phi_{\text{local}}(t) \rangle,$$

- And leakage

$$p_{\text{leak}}^{\text{local}}(t) = 1 - p_{\text{in}}^{\text{local}}(t).$$

From backend:

- Approximate P using dominant basis states $\mathcal{B}_{\mathcal{M}}$ and compute:

$$p_{\text{in, basis}}^{\text{backend}}(t) = \sum_{z \in \mathcal{B}_{\mathcal{M}}} P_{\text{backend}}(z, t).$$

Agreement at the manifold level means:

- Similar leakage behaviour:

- Backend leakage curve tracks the local leakage curve up to noise and damping.
- Similar confinement: dynamics stays in the same “region” of Hilbert space for both backend and local.

5.2 Basis-structure indicators

Because you have **dominant basis-state structure** for each eigenstate:

- You know which bitstrings “belong” to which part of the spectrum.
- You can aggregate probabilities into “buckets” (e.g., low-energy dominant patterns vs high-energy patterns) both locally and on backend.

Agreement means:

- Backend probability mass flows between these buckets over time in the same way that the local model predicts.
- Backend does not spuriously populate bitstrings that should be very unlikely according to the local Hamiltonian (beyond what noise and readout errors explain).

This gives a very **interpretable** picture of backend vs local behaviour, rather than only abstract fidelity metrics.

6. Agreement scales: short-time vs long-time behaviour

Typically, you will see different regimes:

Short times (few Trotter steps)

- Trotter error is small.
- Hardware noise has had little time to accumulate.
- Backend vs local agreement should be strongest:
 - Oscillation frequencies match,
 - Amplitudes close,
 - Leakage small and consistent.

Intermediate times

- Trotter error grows with t (for fixed m).
- Decoherence starts to significantly damp oscillations.
- You can still see whether **frequencies** (eigenvalue differences) match, even if amplitudes are reduced.
- This is a good regime to test the **spectral structure** of my Hamiltonian.

Long times

- Noise washes out structure; distribution drifts towards something closer to uniform or readout-biased.
- Backend vs local agreement typically collapses.
- This regime is more about probing noise characteristics than about validating H .

So “Backend vs Local Agreement” is not just yes/no; it is **time-dependent**, and v0.5 should explicitly acknowledge that you only expect meaningful agreement up to a certain time scale set by T1/T2 and gate errors.

7. Practical metrics you can explicitly mention in v0.5

For documentation / methods, you can define concrete agreement metrics like:

- **Observable error:**

$$\Delta_O(t) = |\langle O(t) \rangle_{\text{backend}} - \langle O(t) \rangle_{\text{local}}|.$$

- **Distribution error:**

$$D_{\text{TV}}(t) = \frac{1}{2} \sum_z |P_{\text{backend}}(z, t) - P_{\text{local}}(z, t)|.$$

- **Manifold leakage discrepancy:**

$$\Delta_{\text{leak}}(t) = |p_{\text{leak}}^{\text{backend}}(t) - p_{\text{leak}}^{\text{local}}(t)|.$$

- **Time-averaged agreement** over a time window $[0, T]$:

$$\bar{\Delta}_O = \frac{1}{T} \int_0^T \Delta_O(t) dt \text{(or discrete sum).}$$

You can then specify practical thresholds:

- $\Delta_O(t)$ below some tolerance for all $t \leq T_{\text{trust}}$,
- $D_{\text{TV}}(t)$ less than a value consistent with shot noise,
- Etc.

Those thresholds define my “**trust window**” where backend data can be safely interpreted via my local model.

8. Conceptual summary in v0.5 language

You can phrase “Backend vs Local Agreement” roughly like this in my v0.5 framework:

- **Local model:** A noiseless classical simulation of the same Pauli Hamiltonian and Trotter sequence used on the IBM backend, including full spectral data (eigenvalues, eigenvectors, manifolds, dominant basis states).
- **Backend experiment:** Execution of the corresponding Trotter circuits on a real IBM device, with measurement statistics collected over a discrete time grid.
- **Agreement:**
 - At the distribution level: $P_{\text{backend}}(z, t)$ is close to $P_{\text{local}}(z, t)$ in a suitable distance measure.
 - At the observable level: $\langle O(t) \rangle_{\text{backend}}$ tracks $\langle O(t) \rangle_{\text{local}}$, matching both frequencies and qualitative envelopes.
 - At the manifold level: backend dynamics respects the same near-degenerate manifolds and dominant basis-state regions (similar subspace leakage behaviour).
- **Disagreement** stems from:
 - Model mismatch (real device vs idealized Hamiltonian),
 - Trotter approximation (finite step number / order),
 - Hardware noise and readout errors,
 - Finite sampling.
- **Use:** Backend vs Local Agreement is a diagnostic:
 - To validate that the hardware is implementing my intended Hamiltonian (within a certain time/energy window),
 - To decide up to which times t my quantum dynamics experiments can be interpreted in terms of the v0.3/v0.4 spectral structures (manifolds, degeneracies, dominant basis states),
 - And to tune Trotter parameters and circuit design until agreement is as good as the hardware allows.