AI-Driven Discovery of Physics-Native Quantum Information Units
Version 1.1 — December 2025

Abstract

This document presents Version 1.1 of the research concept proposing that quantum information units may naturally emerge from the symmetry, degeneracy, or low-energy structure of physical Hamiltonians.

The goal is to explore whether modern AI systems can identify such physics-native information units, potentially reducing the overhead associated with conventional, engineered qubits. This version strengthens the conceptual motivation, provides examples of relevant Hamiltonians, adds related work context, and expands the system architecture description.

1. Introduction

Contemporary qubit designs rely on engineered logical abstractions that impose large physical-to-logical overheads. Recent improvements in error-correcting codes and decoding strategies suggest that these overheads may reflect not intrinsic physical limitations, but rather a mismatch between the abstractions we impose and the information-preserving structures present in the underlying physics.

Motivated by Feynman's original intuition—that computation should follow the natural structure of quantum mechanics—this work explores whether stable, low-noise information-bearing subspaces can be discovered using AI-assisted analysis of Hamiltonian landscapes.

2. Conceptual Motivation

The central question is whether nature offers information encodings that differ from the qubit model. Rather than defining computational units a priori, we examine whether physics-native encodings arise from:

• symmetry-protected subspaces

• degenerate manifolds

• conserved quantities

• low-energy basins shaped by the Hamiltonian

These structures may exhibit stability without relying on heavy error correction, and could serve as alternative

information units aligned with the system's physical dynamics.

3.Mathematically Elegant vs. Physically Natural Representations

In many areas of quantum computing, the mathematical representation of computation is valued for its elegance: clean tensor products, well-defined gate sets, and compact algebraic descriptions. However, physical systems do not necessarily organize information according to our preferred mathematical structures.

A circuit may look "beautiful" on paper — composed of symmetric patterns of Hadamard, phase, and controlled operations — yet remain physically fragile because the underlying interactions break these imposed symmetries. Conversely, a physically stable structure may appear mathematically irregular or cumbersome, but derives its resilience precisely from the native dynamics of the Hamiltonian.

This tension suggests that mathematical elegance is not a reliable proxy for physical suitability. Physics may preserve information in subspaces that do not resemble qubits, that lack clean algebraic decompositions, or that arise from accidental or non-obvious symmetries. Such structures are often overlooked because they do not match the engineered abstractions commonly used in qubit architectures.

Identifying these physically natural encodings — even when they do not align with standard quantum gate formalisms — may be essential if we wish to develop quantum information units that are stable for reasons intrinsic to the system, rather than enforced through heavy error correction.

4. Example Hamiltonians

To ground the idea, Version 1.1 introduces three illustrative Hamiltonians:

1. Transverse Field Ising Model:

$H = -J \Sigma \sigma^z_i \sigma^z_{i+1} - h \Sigma \sigma^x_i$

Known for symmetry-breaking and stable subspace behavior.

2. Kitaev Chain:

Supports Majorana zero modes and topological degeneracy, a candidate for physics-native encoding.

3. Heisenberg XXZ Model:

Exhibits conservation laws and symmetry structures that influence stability.

These examples serve as potential landscapes for AI-assisted discovery of stable subspaces.

4.1 Emergent Subspace Structure in Random Hamiltonians

Physics-Native Subspace Discovery – Qiskit Prototype v0.1b

4.1.1. Background and Objective

In this document I document the first concrete prototype implementation of the ideas outlined in my whitepaper on physics-native quantum information units.

The central central question is whether small, low-dimensional subspaces with intrinsic stability can emerge directly from the Hamiltonian, without imposing the qubit abstraction a priori. In particular, I explore whether random, Pauli-structured Hamiltonians already contain nearly degenerate energy subspaces that could serve as 'physics-native' information units.

The goal of this first experiment is not to prove superiority over conventional qubits, but simply to demonstrate that:

1. A small Qiskit-based engine can scan a family of Hamiltonians.

2. It can automatically identify near-degenerate eigenvalue pairs.

3. The corresponding eigenstates show non-trivial structure in the computational basis.

This document describes the code, shows the raw output from one seed scan, and outlines possible next steps.

4.1.2. Method: Qiskit Prototype v0.1b

4.1.2.1 Hamiltonian Construction

The Hamiltonians used in this prototype are defined on two qubits and expressed as linear combinations of Pauli strings using Qiskit's SparsePauliOp.

Each Hamiltonian H is constructed as:

$H = \Sigma c_k P_k$

where each P_k is a tensor product of single-qubit Pauli operators chosen from {I, X, Z}, and $c_k \in \mathbb{R}$ are random coefficients drawn from a uniform distribution in [-1, 1].

Identity-only strings (e.g., 'II') are excluded, since they merely shift the overall energy and carry no structural information.

4.1.2.2 Eigenvalue Analysis and Near-Degeneracy Criterion

For each Hamiltonian, the full 4×4 matrix representation is obtained via hamiltonian_op.to_matrix().

I then compute its eigenvalues and eigenvectors using numpy.linalg.eigh. This yields:

$H |\psi_i\rangle = E_i |\psi_i\rangle$ for $i = 0, 1, 2, 3$.

To identify potentially 'stable' or structurally interesting subspaces, I define a near-degeneracy criterion:

$|E_i - E_j| < \varepsilon$

with $\varepsilon = 0.05$. Any pair (i,j) satisfying this condition is reported as a candidate two-dimensional subspace.

4.1.2.3 Implementation Details

The prototype is implemented in Python using Qiskit. The core functions are:

– build_random_hamiltonian_sparse(...)

– find_near_degenerate_pairs(...)

– print_statevector(...)

– and a simple main loop scanning seeds 0–49.

Insert full code here as needed.

4.1.3. Experimental Run and Output (Seed Scan 0–49)

For this first documented run, I scanned seeds from 0 to 49 with:

– n_qubits = 2

– num_terms = 4

– tolerance = 0.05

The program reported several seeds with near-degenerate eigenvalue pairs. Insert selected example outputs here (e.g., seeds 0, 23, 31).

4.1.4. Interpretation of Results

The seed scan confirms that even in a small, randomised family of 2-qubit Hamiltonians built from {I, X, Z}, near-degenerate energy levels appear frequently.

In many cases (e.g., seeds 0, 2, 9, 14, 26, 30, 31, 34), the spectrum exhibits exact or near-exact two-fold degeneracies at both the lowest and highest energy levels. The corresponding eigenstates are not arbitrary; they show recognizable structure in the computational basis.

This suggests that certain Hamiltonians naturally carve out two-dimensional subspaces that are energetically isolated and internally structured.

4.1.5. Proposed Next Steps

1. Systematic logging: Store all Hamiltonians and their near-degenerate pairs in CSV format.

2. Backend Integration: Use Qiskit Estimator on simulators and real IBM Quantum hardware.

3. Scaling to 3 qubits.

4. Noise and stability tests.

Physics-Native Subspace Discovery – Qiskit Prototype v0.3

4.2.1. Background and Objective

This section extends earlier versions using identical structure.

4.2.2. Method: Qiskit Prototype v0.3

4.2.2.1 Hamiltonian Construction

Same method extended to 8 qubits.

4.2.2.2 Eigenvalue Analysis and Near-Degeneracy Criterion

Tolerance unchanged; matrix dimension increased.

4.2.2.3 Implementation Details

Structure identical to v0.1b and v0.2.

4.2.2.4 Code Listing (v0.3)

```
import numpy as np
from qiskit.quantum_info import SparsePauliOp, Statevector

# ---------------------------------------------------------
# 1. Byg en simpel "legetøjs-Hamiltonian" i Qiskit-format
# ---------------------------------------------------------

def build_random_hamiltonian_sparse(n_qubits=2, num_terms=4, seed=None):
    """
    Returnerer en lille tilfældig Hamiltonian som SparsePauliOp.
    Vi bruger kun I, X, Z for at holde det simpelt.
    """
    rng = np.random.default_rng(seed)

    pauli_letters = ["I", "X", "Z"]
    pauli_terms = []

    for _ in range(num_terms):
        # Byg fx "ZX", "IZ", "XI" osv.
        label = "".join(rng.choice(pauli_letters) for _ in range(n_qubits))

        # Undgå ren identitet "II"
        if set(label) == {"I"}:
            continue

        coeff = rng.uniform(-1.0, 1.0)
        pauli_terms.append((label, coeff))
```

```python
    # Hvis alt blev filtreret væk:
    if not pauli_terms:
        pauli_terms = [("Z" + "I"*(n_qubits-1), 1.0)]

    labels = [p for p, c in pauli_terms]
    coeffs = [c for p, c in pauli_terms]
    H = SparsePauliOp.from_list(list(zip(labels, coeffs)))
    return H


# ---------------------------------------------------------
# 2. Find eigenvalues/eigenvectors + near-degenerate pairs
# ---------------------------------------------------------

def find_near_degenerate_pairs(hamiltonian_op, tolerance=0.05):
    """
    Finder par af egenværdier, der ligger tæt på hinanden (|Ei - Ej| < tolerance).
    Returnerer (evals, evecs, pairs) hvor pairs er liste af (i, j, Ei, Ej).
    """
    H_mat = hamiltonian_op.to_matrix()
    evals, evecs = np.linalg.eigh(H_mat)

    pairs = []
    for i in range(len(evals)):
        for j in range(i + 1, len(evals)):
            if abs(evals[i] - evals[j]) < tolerance:
                pairs.append((i, j, evals[i], evals[j]))

    return evals, evecs, pairs


# ---------------------------------------------------------
# 3. Helper: vis eigenstates i basis |00>, |01>, ...
# ---------------------------------------------------------

def print_statevector(evec, n_qubits=2, max_terms=4):
    """
    Viser de største amplituder i en eigenvektor.
    """
    sv = Statevector(evec).data  # numpy-array

    idx_sorted = np.argsort(-np.abs(sv))
```

```
    print("  Dominant basis-states:")
    shown = 0
    for idx in idx_sorted:
        amp = sv[idx]
        if np.abs(amp) < 1e-3:
            continue

        bitstring = format(idx, f"0{n_qubits}b")
        print(f"    |{bitstring}> : amplitude {amp:.3f}")
        shown += 1
        if shown >= max_terms:
            break


# ---------------------------------------------------------
# 4. MAIN – seed-scan
# ---------------------------------------------------------

if __name__ == "__main__":
    n_qubits = 2
    num_terms = 4
    tolerance = 0.05
    max_seed = 50  # prøv seeds 0..49

    print(f"Scanner {max_seed} tilfældige Hamiltonians for near-degenerate par...")
    found_any = False

    for seed in range(max_seed):
        H = build_random_hamiltonian_sparse(
            n_qubits=n_qubits,
            num_terms=num_terms,
            seed=seed
        )
        evals, evecs, pairs = find_near_degenerate_pairs(H, tolerance=tolerance)

        if pairs:
            found_any = True
            print("\n========================================")
            print(f"Seed {seed} gav near-degenerate par")
            print("Hamiltonian:")
            print(H)
            print("Eigenvalues:")
            for i, ev in enumerate(evals):
```

```
        print(f"  {i}: {ev:.4f}")

      print("\nCandidate near-degenerate pairs:")
      for (i, j, ei, ej) in pairs:
        print(f"  Pair ({i}, {j}) with energies {ei:.4f}, {ej:.4f}")
        print("   State", i)
        print_statevector(evecs[:, i], n_qubits=n_qubits)
        print("   State", j)
        print_statevector(evecs[:, j], n_qubits=n_qubits)
        print()

  if not found_any:
    print("\nIngen near-degenerate par fundet for nogen seeds "
       f"med tolerance = {tolerance}. "
       "Du kan prøve at øge tolerance lidt (fx 0.1).")
```

4.2.3 Experimental Results (v0.3)

Overview

This section summarizes the experimental findings from the v0.3 prototype, which scanned multiple randomly generated Hamiltonians to identify degeneracies and near-degenerate eigenpairs.

Seeds Scanned

A total of 50 random seeds were scanned in this prototype.

Degeneracies Found

Several seeds produced clear or near-clear degeneracies. These included Seeds 0, 2, 9, 14, 23, 25, 26, 30, 31, 34, and 35.

Typical Eigenstate Patterns

Across the identified degeneracies, eigenstates consistently exhibited structured combinations of computational-basis states. Often, only two or three basis states dominated each eigenvector, forming coherent low-dimensional subspaces.

Representative Outputs

Examples include:

• Seed 0: Two exact degenerate pairs with clear |01>, |00> and |11>, |10> patterns.

• Seed 2: Symmetric amplitude distributions across |00>, |01>, |10>, |11>.

• Seed 9: Balanced multi-state structures showing consistent subspace formation.

Interpretation

These results demonstrate that even randomly generated Hamiltonians can exhibit well-structured degeneracies. The emergence of stable low-dimensional subspaces supports the broader investigation into physics-native quantum information units.

4.3 Interpretation of Results

Patterns observed across expanded Hilbert space.

4.5. Proposed Next Steps (toward v0.4)

Integration with backend; stability metrics.

5. System Architecture

The proposed pipeline consists of:

5.1 Hamiltonian Library

A structured set of analytic and synthetic Hamiltonians chosen for their symmetry and noise-relevant properties.

5.2 AI Discovery Engine

Neural architectures trained to detect approximate symmetries, conserved quantities, and low-decoherence regions.

5.3 Dynamics Simulator

Time evolution tools that assess stability under realistic noise models.

5.4 Stability Scoring

Metrics comparing candidate subspaces with standard qubits, emphasizing error resilience.

5.5 Realization Filter

A mapping from candidate encodings to feasible physical platforms (superconducting, trapped ions, photonics, etc.).

6. Related Work

This work aligns with emerging research in:

• quantum optimal control

• variational subspace discovery

• symmetry-protected quantum information (SPT phases)

• AI-assisted Hamiltonian learning

• topological computation

These areas suggest that alternative information encodings may be physically meaningful and computationally useful.

7. Limitations

This work remains conceptual. Key limitations include:

• absence of numerical results (future work)

• no guarantee that discovered subspaces will outperform qubits

• uncertainties in physical realizability

• potentially high computational cost in large Hamiltonian spaces

Nonetheless, the conceptual framework motivates further exploration.

8. Discussion and Outlook

If physics-native information units exist and can be systematically identified, the implications may include:

• reduced qubit overhead

• architectures aligned with underlying physical structure

• new pathways for error resilience

• a return to Feynman's intuition about computation following physics

Future work will extend the Hamiltonian library, develop prototype AI models, and perform stability tests.

8.1 Future Work Roadmap

The present whitepaper represents an initial exploration of physics-native information units emerging from Hamiltonian structure. Several directions now naturally follow from the numerical evidence and prototype implementations presented above:

(1) Expanded Hamiltonian Sampling

Future versions will extend the random-Hamiltonian scan from two-qubit systems to larger Hilbert spaces, enabling statistical characterization of how often degenerate and near-degenerate subspaces arise as a function of qubit count, coupling structure, and operator distribution.

(2) Subspace Stability Analysis

A next step is to evaluate the robustness of identified subspaces under perturbations, noise channels, and drift in Hamiltonian coefficients. This includes constructing projectors such as:

$$P = \sum_k |\psi_k\rangle\langle\psi_k|$$

and testing their invariance properties under decoherence models.

(3) Backend Validation on Real Quantum Hardware

The Qiskit prototype will be extended to execute selected Hamiltonians and projectors on cloud-accessible quantum processors. This will provide empirical insight into how physics-native subspaces behave under real device noise and calibration cycles.

(4) Operator-Algebraic Encoding Framework

The long-term goal is to explore whether the subspaces identified by the AI-assisted pipeline form the basis for stable logical units. This includes studying the algebra of operators acting within the subspaces and evaluating whether they support gate sets with low induced noise.

(5) Integration into the AI-First Architecture (Version 2.0)

The next major version of the system will integrate multi-qubit Hamiltonian sampling, degeneracy detection, projector synthesis, and backend evaluation into a unified pipeline capable of proposing, evaluating, and ranking candidate physics-native encodings.

To be continued in Version 1.2, where larger-qubit analyses, projector-based subspace encoding, and backend validation will be added.

References

[1] Preskill, J. (2018). Quantum Computing in the NISQ era.

[2] Kitaev, A. (2001). Unpaired Majorana Fermions.

[3] Wen, X.-G. (2017). Topological Order.

[4] Feynman, R. P. (1982). Simulating Physics with Computers.

[5] Various authors on quantum optimal control and symmetry-protected subspaces.