This is a pre-print of chapter 10 (pp 107-121), a featured contribution to the monograph edited by Oswaldo Zapata, PhD *A Portrait of Quantum Technologies in Finance,* The Quantum Finance Boardroom (2026).

A Dance of the Blind Puppeteer: The Interplay Between a Classical Optimizer and the Hilbert Space

Jacob L. Cybulski[1,2] [0000-0002-9061-9389]

[1]Enquanted, Melbourne, Australia
jacob@enquanted.com
http://jacobcybulski.com/

[2]Deakin University, School of IT, Melbourne, Australia
https://experts.deakin.edu.au/173-jacob-cybulski

April 2026

Abstract

This chapter explores the intricate training dynamics of quantum machine learning models, framed as an interplay between classical optimization and the vast geometry of the Hilbert space. We move beyond the common trope of Hilbert space "vastness," focusing instead on a geometry sculpted by quantum entanglement and operational constraints. We elucidate the "Blind Puppeteer" problem: the classical optimizer's struggle to refine classical parameters without insight into their quantum representation or their role in evolving qubit states. From a purely geometric perspective, we analyze how the high dimensionality of both the parameter and Hilbert spaces culminates in training pathologies like Barren Plateaus, Orthonormal Desert, and the Measurement Sparsity. Ultimately, we argue for the necessity of "quantum-aware" optimizers—tools capable of navigating the subtle curvature of the quantum manifold that standard gradient-based methods ignore.

Keywords: Quantum Machine Learning, Quantum Model Optimization, Hilbert Space, Curse of Dimensionality, Barren Plateaus, Orthonormal Desert, Measurement Sparsity, Quantum Natural Gradient

1 Introduction

This chapter presents a concise synthesis of selected concepts, methods, and procedural frameworks in Quantum Machine Learning (QML), which often represent substantial entry barriers for researchers and practitioners in applied Quantum Computing (QC).

The motivation for using QML as a problem-solving toolkit is typically two-fold. First, it can directly model quantum phenomena with inherent nondeterminism, high-dimensional feature spaces, and complexity (e.g., quantum chemistry or quantum physics). Second, it applies Quantum Mechanics as a lens for complex classical problems characterized by imprecision, large volumes of poor quality or aging data, and dissatisfaction with deterministic or stochastic

methods—particularly in quantum finance or optimization, where one already contends with complex combinatorial landscapes. In this chapter, we focus on the latter category of problems and potential QML solutions, which commonly arise in business computing, and in particular logistics, supply chain management, and finance.

To support the adoption of QML for business problem solving, foundational quantum computing concepts must be complemented with calculus, data science, information geometry, and an understanding of how classical computation interacts with quantum phenomena. This chapter provides intuition and a conceptual framework for core QML ideas at the boundary of the classical and quantum realms. We focus on developing predictive and decision-support quantum models trained with classical optimization algorithms that, while robust in classical domains, largely ignore the unique geometry of quantum effects. In this context, we portray such an optimizer as a *blind puppeteer*, pulling the marionette's strings to make it dance, guided only by the intermittent cheers of the audience as the sole, indirect feedback on his artistry.

2 Strings: PQC Parameters

Quantum circuits consist of qubits, operations, and measurements. Executing a circuit produces a quantum state that evolves under these operations. A standard circuit is static: it hardcodes classical data and parameters, so any change requires building a new circuit. Such circuits cannot be easily adapted or optimized for different analytic goals or data. However, a Parameterized Quantum Circuit (PQC) provides a template that separates the circuit structure and function from its parameters (Figure 1).

PQCs can be designed to fulfill specific analytic objectives, e.g. to generate predictions or decisions based on data representing past events. Such PQCs typically consist of the following functional sections:

- *Feature map* embedding classical data in the PQC's input parameters associated with operations responsible for setting the model's initial quantum state;
- *Ansatz* consisting of qubit entangling blocks and blocks of rotational operations parameterized with circuit weights, which could be trained to evolve the initial circuit state into a target state;
- *Measurements* responsible for decoding the target state of the executing circuit into a classical data form, which is interpreted and returned as output.

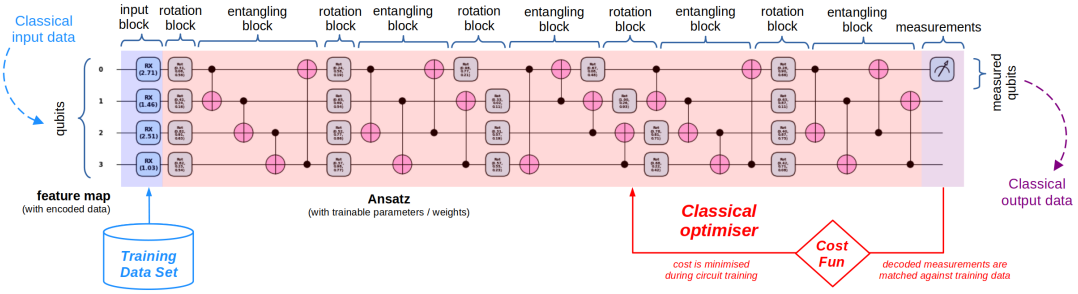


Figure 1: Parameterized quantum circuit trained with variational quantum algorithm.

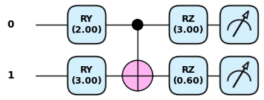
The components of these sections may be intermixed, e.g. data embedding blocks can be interwoven with encoding and trainable ansatz blocks (known as data reuploading), or measurements can appear in the midst of the ansatz (known as mid-circuit measurement).

PQC training typically uses a Variational Quantum Algorithm (VQA) (Figure 1) that relies on a classical optimizer, which is equipped with a classical training dataset and a cost function, to iteratively arrive at the optimum values of the circuit weight parameters. The role of a cost function is to aggregate the results of the associated loss function, which calculates the circuit execution error. In this process, the optimizer acts completely unaware of any quantum processes prescribed by the quantum circuit. However, it is aware and in full control of the classical elements of the circuit, i.e. its input and output values, the ansatz weights, and the result of circuit execution.

The classical circuit elements controlled by the "quantum-blind" optimizer are considered the puppet strings.

3 Stage and Puppet: Hilbert Space and Manifold

To understand quantum machine learning, one must first visualize the "stage" upon which the quantum model performs—the Hilbert space [4]. While we often speak of the Hilbert space as the arena of quantum states, it must be formally treated as a *Projective Hilbert Space* to be physically meaningful. This is achieved by adding an equivalence relation that treats all vectors differing only by a complex scalar as the same state, effectively collapsing the linear space into a multi-dimensional complex projective surface, referred to as the *manifold*. Unlike the familiar three-dimensional Euclidean space of classical physics, the state space of a quantum system is curved, and hence not measured in straight lines, but by the "overlap" or *fidelity* between quantum states.




```
@qml.qnode(dev)
def project_to_3d_v1(params):
    qml.RY(params[0], wires=0)
    qml.RY(params[1], wires=1)
    qml.CNOT(wires=[0, 1])
    qml.RZ(0.5 * params[0] * params[1], wires=0)
    qml.RZ(0.3 * params[0], wires=1)
    return (
        qml.expval(qml.PauliX(0)),
        qml.expval(qml.PauliZ(1)),
        qml.expval(qml.PauliZ(0) @ qml.PauliZ(1))
    )
```

Figure 2: Sample quantum circuit and its implementation in PennyLane.

A PQC acts as a map that translates classical numbers—the parameters—into specific quantum state configurations in the Hilbert space. However, this mapping is rarely one-to-one or uniform. Because of quantum entanglement, some PQC states become inseparable, which creates a *manifold*—a lower-dimensional surface within the massive Hilbert space—upon which the model is forced to reside. The shape of this manifold dictates whether the model can reach the "ideal" state required to solve a problem or whether it is trapped in a geometric dead-end. The geometry of the manifold is constrained by the mathematical laws of superposition, entanglement, and the *Fubini-Study metric* (FS) [5].

While *fidelity* measures state similarity, the *Fubini-Study metric* defines the intrinsic geometry of the Hilbert space. By characterizing how a quantum output state changes under infinitesimal parameter shifts, it sets the local curvature of the training surface, "warping" Euclidean parameter space into regions of high sensitivity or stagnation. FS provides the mathematical basis of the *Quantum Fisher Information Matrix* responsible for mapping the

Table 1: Layer-by-layer state evolution and observable projections (cf. Figure 2). Notation: $p_0 \equiv \texttt{params}[0]$, $p_1 \equiv \texttt{params}[1]$, $c_i \equiv \cos(p_i/2)$, $s_i \equiv \sin(p_i/2)$, $\phi_0 \equiv p_0 p_1/4$, $\phi_1 \equiv 0.15\, p_0$.

Step	**State / Observables**	**Comment**
1	$\lvert\psi_1\rangle = c_0\lvert 00\rangle + s_0\lvert 10\rangle$	$R_Y(p_0)$ rotates qubit 0 into superposition.
2	$\lvert\psi_2\rangle = c_0c_1\lvert 00\rangle + c_0s_1\lvert 01\rangle + s_0c_1\lvert 10\rangle + s_0s_1\lvert 11\rangle$	$R_Y(p_1)$ rotates qubit 1; state remains separable.
3	$\lvert\psi_3\rangle = c_0c_1\lvert 00\rangle + c_0s_1\lvert 01\rangle + s_0s_1\lvert 10\rangle + s_0c_1\lvert 11\rangle$	CNOT swaps $\lvert 10\rangle \leftrightarrow \lvert 11\rangle$; generates entanglement.
4	$\lvert\psi_4\rangle = e^{-i\phi_0}(c_0c_1\lvert 00\rangle + c_0s_1\lvert 01\rangle) + e^{+i\phi_0}(s_0s_1\lvert 10\rangle + s_0c_1\lvert 11\rangle)$	$R_Z(\phi_0/2)$ on qubit 0 applies control-dependent phase, adds nonlinear/interactive warping.
5	$\lvert\psi_{\text{fin}}\rangle = c_0c_1e^{-i(\phi_0+\phi_1)}\lvert 00\rangle + c_0s_1e^{-i(\phi_0-\phi_1)}\lvert 01\rangle + s_0s_1e^{+i(\phi_0-\phi_1)}\lvert 10\rangle + s_0c_1e^{+i(\phi_0+\phi_1)}\lvert 11\rangle$	$R_Z(\phi_1)$ on qubit 1 adds independent phase, linear/gauge-like warping; final state.
6	$\langle Z_1\rangle = \cos p_0 \cos p_1, \langle Z_0Z_1\rangle = \cos p_1$ $\langle X_0\rangle = \sin p_0 \sin p_1 \cos(p_0p_1/2)$	Expectation values: Z-terms phase-invariant; $\langle X_0\rangle$ encodes nonlinear interference, it contains $\cos(p_0p_1/2)$ because the R_Z rotation "drags" the state around the Z-axis, modulating the projection onto the X-axis.

curvature of the optimization landscape, hidden from the classical optimizer, but which it nevertheless unknowingly traverses.

A quantum model manifold can be derived mathematically from its circuit structure, where the unitary parameterization defines the state evolution, and consequently the model geometry, which exists independently of the circuit execution (see Table 2). As illustrated in Table 1, a quantum circuit (Figure 2) provides sufficient details to predict its step-by-step state evolution. Furthermore, we can predict the warping of the circuit manifold in response to entanglement operations and nonlinearities resulting from R_Z rotations, which do not alter quantum circuit measurements (in Z basis), but which accumulate bending the model manifold (in X axis that is non-commutative with Z rotations) as parameters change.

Alternatively, we can rely on a quantum processor or simulator to empirically approximate the manifold by sampling the circuit expectation values from finite-shot measurements (Figure 3a). In such an approach, training data are necessary to drive the geometric reconstruction process. However, data themselves do not "create" the manifold. Instead, data values act as coordinates that select specific trajectories through the manifold (e.g. see state evolution paths in Figure 3a). Apart from entanglements, which are the main agents of shaping the manifold geometry, the ansatz weight parameters are also capable of adjusting the manifold's curvature.

In practical quantum computing, as opposed to the purely mathematical conceptualization

Table 2: What aspects contribute to manifold creation

Concept	What it is	Does it define the manifold?
Circuit Ansatz Parameterization	The mapping $(\theta, x) \mapsto U(\theta, x)\,\|0\rangle$	**Yes.** Mathematically defines the intrinsic manifold $\mathcal{M} = \{\|\psi(\theta, x)\rangle\}$.
Input Encoding	Data x enters as gate parameters, e.g., $R_Y(x_i)$	**Yes.** Data often defines the input state location on the manifold $\mathcal{M}$, selecting specific trajectories or submanifolds.
Ansatz Weights	Weights θ also enter as gate parameters, e.g., $R_Z(\theta_{i1}), R_Y(\theta_{i2}), R_Z(\theta_{i3})$	**Yes.** Weights define the transformation/rotation of the manifold $\mathcal{M}$.
Execution and Multiple Shots	Repeated measurements to estimate $\langle O \rangle$	**No.** Shots *sample* the manifold; they do not create it. Finite shots introduce statistical noise.
Classical Optimizer	Updates θ to minimize a scalar cost function	**No.** Navigates a cost landscape over $\mathcal{M}$, without knowledge of the underlying quantum geometry.

of the Hilbert space, due to decoherence (perturbations to the phase) and noise (perturbations to the magnitude of density matrix elements), data points sampled with NISQ quantum machines do not fall precisely onto the mathematically idealized manifold surface (of pure states). Instead, the empirical manifold forms a cloud of quantum state locations (of mixed states).

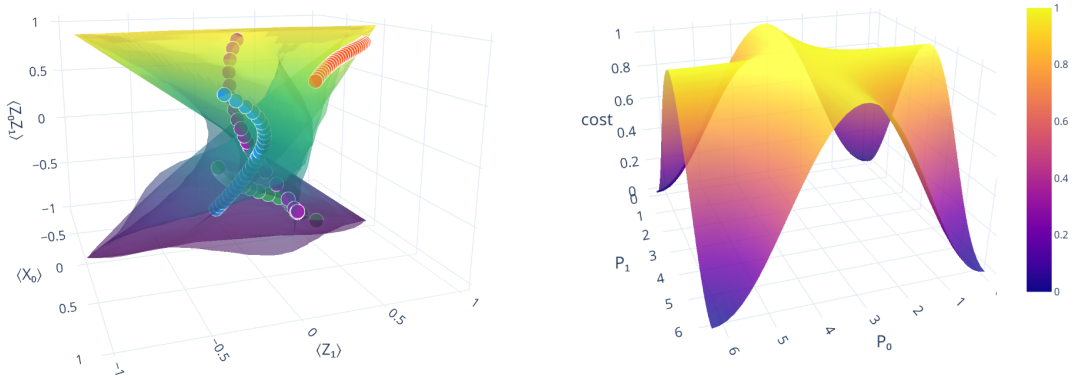
In the world of quantum puppeteers, the Hilbert space is the stage for the performance of the manifold—the puppet.

4 Blind Puppeteer: Classical Optimizer

The actual training of a quantum model is conducted by a classical optimizer – a mathematical algorithm, which adopts the VQA approach but which remains entirely oblivious to the quantum nature of the system it is tuning (Figure 3a).

The optimizer iteratively creates static PQC instances, each with different input samples from the training dataset and different weight parameters. It treats each PQC instance as a black box that, when executed multiple times, returns an approximate output for the chosen inputs. This approximation is compared with the target outputs using a loss function (as implied by the cost function), yielding a single error value, or *loss*. Aggregating these losses over the entire training dataset gives the overall *cost* of a given model parameterization. All possible instantiations of the PQC parameters can be associated with their respective costs, thus forming the *cost landscape* (Figure 3b). Only fragments of the cost landscape are computed to reduce computational costs.

The classical optimizer cannot see qubits, interference, or entanglement, it can only see the cost landscape. Its primary task is to efficiently navigate this landscape to find the geometrically lowest point in the landscape—and consequently the lowest cost and indirectly the optimum circuit parameterization. During its search for the optimum, at each iteration the optimizer



(a) Warped circuit manifold with four selected state evolution paths

(b) Optimization cost landscape, which is lacking any details of the model's quantum geometry

Figure 3: The circuit manifold and its cost landscape

is moving blindly through the parameter space by relying entirely on the local surface shape. Optimizers such as SGD and Adam rely on the landscape slope, or *gradient*, to decide the next improvement to the parameter set. However, gradient-free optimizers, such as Nelder-Mead or COBYLA, also suffer from the same blindness; they simply touch the surface for a lower point rather than measuring the pitch of the slope.

The optimizer is the blind puppeteer, sightlessly animating its high-dimensional parameter-strings in search of the optimum dance steps, listening to the erratic clapping of the audience as the noisy feedback for his artistry.

5 Too Many Strings: The Curse of Dimensionality

The "Curse of Dimensionality" often turns a manageable optimization problem into an intractable one as the number of its parameters or qubits increases, raising the tension between model expressivity (ability to represent data in quantum space) and trainability (capacity to learn and generalize with optimization efficiency) [1]. Higher dimensionality may be required to give the circuit enough expressivity to learn the patterns of a complex dataset, but this typically harms the optimizer effectiveness, and thus the model trainability. To understand why a classical optimizer faces difficulties when training large quantum models, we must look at the geometry of the spaces it is trying to navigate.

Consider the previously discussed quantum model (Figure 2). The representations of its state space (Figure 3a) and its cost landscape spanning the parameter space (Figure 3b) are very different. In this case, the shape of the state space is folded and complex, and the shape of the cost landscape is simple and smooth. However, when their respective dimensions become very high, both shapes will display the same unusual characteristics and problems.

In the explanation of these problems, we will use a well-known concept of a multidimensional unit-size n-ball as a proxy for the parameter space as well as the Hilbert state space—the n-ball principles apply to any high-dimensional space. By analyzing the high-dimensional n-ball properties we encounter a surprising series of paradoxes (cf. Figures 4, 5 and 6).

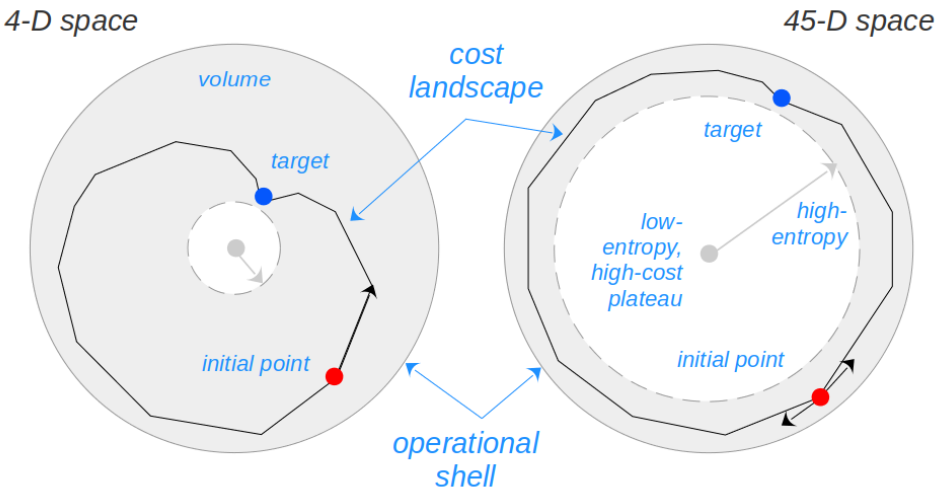


Figure 4: 2D metaphor of a 4-dimensional n-ball (left), and 45-D n-ball with a *Barren Plateau (BP)* (right). Note how volume (grey) in n-ball shrinks with the number of dimensions.

The Shrinking n-Ball Volume Paradox. In our three-dimensional world, we expect that adding dimensions increases the space available for a solution. However, the volume of n-dimensional n-ball with radius R=1 follows a counter-intuitive trajectory defined by the equation:

$$V_n(R) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} R^n, \quad \textit{and} \quad V_n(1) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} \quad \textit{for } R = 1 \tag{1}$$

where $\Gamma(z)$ is a *Gamma function*, an extension of the factorial function to complex numbers, which is defined as follows:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt, \quad \text{for } \operatorname{Re}(z) > 0 \tag{2}$$

As the dimensionality n increases, the volume $V_n(1)$ does not grow indefinitely. It reaches a maximum at $n \approx 5.25$ and then begins an exponential collapse toward zero. By the time we reach the 45-dimensional space (figuratively represented in Figure 4), the total volume of the parameter space has effectively vanished. The space itself does not become empty, instead the mathematical definition of its volume in high dimensions diminishes the space "interior."

The Thinning-and-Expanding n-Ball Shell Paradox. Perhaps the most damaging property for an optimizer is the *concentration of volume*. In high dimensions, the volume of the n-ball is not evenly distributed; instead, it migrates almost entirely to a thin layer near the surface (Figure 4). The volume contained within a surface shell of thickness $\epsilon \in [0, 1]$ is defined as the difference in volume of the n-balls of radius 1 and $(1 - \epsilon)$, which can be shown to be:

$$V_{shell} = V_{total}\,[1 - (1 - \epsilon)^n] \tag{3}$$

As $n \to \infty$, the term $(1 - \epsilon)^n$ vanishes. For example:

- In 3D ($n = 3$): The outer 1% of the radius contains only $\sim$2.9% of the volume.

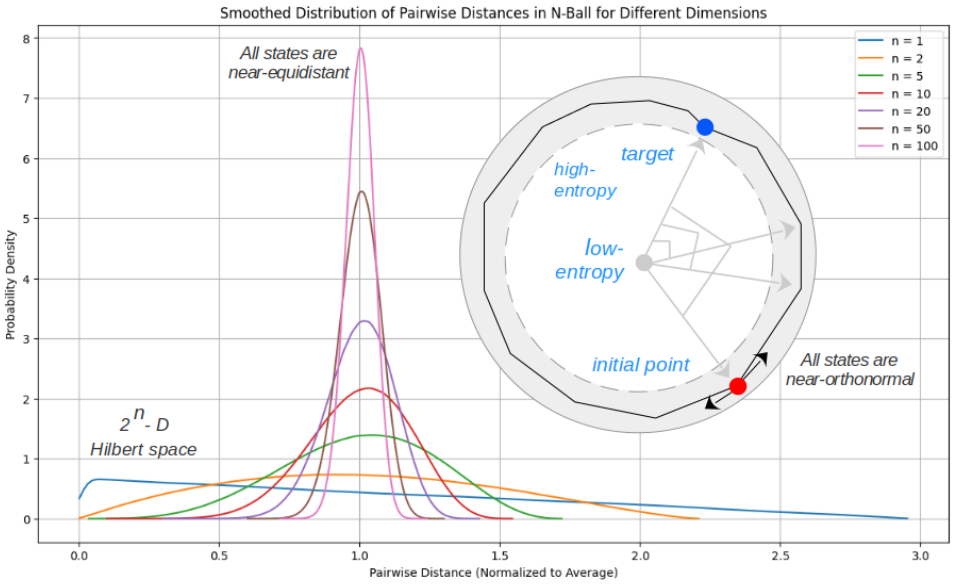


Figure 5: Manifestation of an *Orthonormal Desert*. Note how distribution of pairwise distances between states within an n-ball concentrates around the mean as the dimension increases.

- In 45-D ($n = 45$): The outer 1% of the radius contains $\sim$36.4% of the volume.
- In 500-D ($n = 500$): The outer 1% of the radius contains $\sim$99.3% of the volume.

In our 3D intuition, the outermost 1% of a sphere's radius is an insignificant sliver, containing less than 3% of its volume. However, as we move to a 45-dimensional model, this same 1% shell expands to claim more than 36% of the total searchable space.

In the n-ball proxy of the Hilbert space, n represents the dimension of the state space ($2^{\#qubits}$), illustrating how a linear increase in physical qubits leads to an exponential 'thinning' of the interior; e.g. at 9 qubits, the model state space already exceeds 500 dimensions ($2^9 = 512$). In the parameter space, a circuit of 19 qubits with 9 rotational blocks of 3 rotational operations each (R_z, R_y, R_z) exceeds 500 dimensions ($19 \times 9 \times 3 = 513$). After that point, the volume rapidly migrates toward the surface, leaving the interior hollow.

For the optimizer, this creates a serious difficulty. Any movement the optimizer makes is statistically forced to be *tangential* to the surface. It is no longer moving "toward" or "away" from anything—it is merely drifting around a very thin high-dimensional shell.

Gradient Extinction Paradox. Because the "interior" has vanished and all content is held in a thin surface shell, the "slope" between an arbitrary pair of points becomes too shallow to detect. This *concentration of gradient* around zero makes any surface within the shell completely flat. In the parameter space, the cost landscape becomes a *Barren Plateau*, i.e. also flat [3] (Figure 4), depriving the classic optimizer of any meaningful gradient to follow.

Orthonormal Desert and Equidistance Paradox. In a curious twist of multi-dimensional mathematics, any two random vectors pointing to the surface of the n-ball are approximately orthogonal to each other (Figure 5, right). This *concentration of angular separation* around orthogonality can be formally justified by examining the dot product of these vectors. Since

in each coordinate dimension, the product of the corresponding components is equally likely to be positive or negative, the contributions from individual dimensions tend to cancel out. Consequently, in a high-dimensional space, the sum of these products converges statistically toward zero, leading to expected near-orthogonality between randomly selected vectors.

This makes the majority of the n-ball volume the *Orthonormal Desert*.

At the same time, it can be shown through computational experiments that a distribution of distances between randomly selected points in high-dimensional n-ball concentrates close to their mean (Figure 5, left). In fact, since all points in the thin n-ball crust shell are equidistant from its center, the radius of the ball is 1 and the points are orthogonal, their pairwise distance must be approximately $\sqrt{2}$, creating an incredible *concentration of distance* in high-dimensional space.

From the classical optimizer's perspective, the Hilbert space is not only very large but also incredibly homogeneous in all directions.

The resulting state space is a highly fractured and directionless void. The classical optimizer is not just stuck in the local minimum, it is trapped on a high-dimensional manifold where the very concept of a path toward the target has been erased by the geometry of the space itself. While the optimization solution exists, the geometry of the n-ball ensures that the optimizer lacks the "volume", "direction" and "smoothness" to navigate the n-ball space leading to it.

In such situations, one of the most deceptive aspects of quantum model training is the appearance of convergence. Because the cost landscape is subject to extreme flattening due to *Barren Plateaus*, the gradient signal often drops below the numerical or statistical noise floor. To a classical optimizer, this lack of signal is indistinguishable from reaching a true minimum. Practitioners may find their models converging quickly to a stable cost value, only to discover that the resulting "optimal" quantum state remains deep within the *Orthonormal Desert*—nearly orthogonal to the target and possessing no actual predictive power.

Entropy Explosion Paradox. Because the radial probability of finding a state at a given distance from the center $P(r) = nr^{n-1}$ vanishes at the origin, the n-ball center is a region of minimal entropy—a unique, highly ordered state that occupies no volume in the high-dimensional landscape. In contrast, the n-ball outer shell is a high-entropy mix of states, containing the vast majority of state configurations. Paradoxically, this leads to *concentration of entropy*, where uncertainty becomes confined to an extremely narrow range of space.

In terms of circuit states, the states that are most likely to occupy the vicinity of the n-ball center are unique, specific, and unlikely to be encountered by the optimizer.

Also, because the volume of the n-ball is concentrated near the surface, a random initialization of the circuit state (the red dot in Figure 4) is statistically likely to land in a narrow but high-entropy (s)hell. The target state (the blue dot in Figure 4) of the lowest cost is also likely to be located in the same high-entropy region, where all states look equally probable and orthogonal, thus making the target a "needle in a high-entropy haystack."

The optimizer begins its journey not only far from the target, but its search is also hampered by the huge volume of competing states packed into a narrow sliver of space—a high-entropy labyrinth where every direction looks the same to a blind puppeteer.

The Statistical Conundrum. While the geometry of the n-ball ensures that the directional signal is mathematically extinguished, measurement difficulties create a separate practical bot-

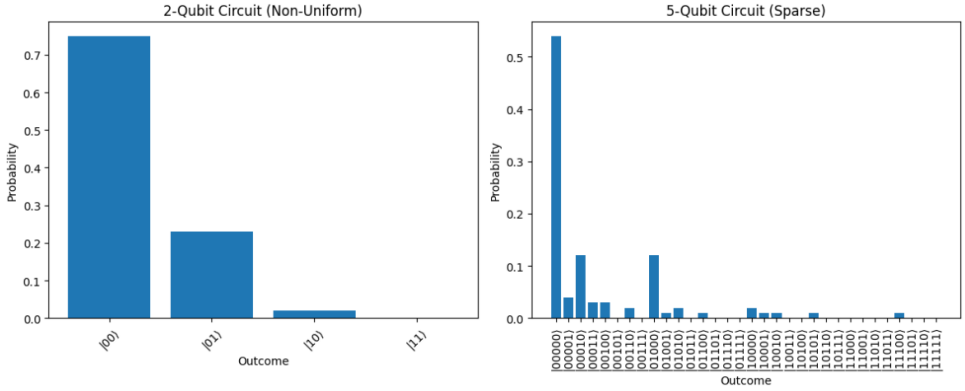


Figure 6: Manifestation of *Measurement Sparsity*. Note the sparse distribution of measurement outcomes. As counts (shots) are low, the calculated probabilities become imprecise.

tleneck. This is not merely a matter of diminishing gradients, but a fundamental issue of statistical resolution [2].

Figure 6 illustrates what happens when the number of qubits (n) increases and the dimension of the outcome space consequently grows exponentially (2^n). In such situations, a fixed number of measurement "shots" must be distributed across a vast array of potential results. In low dimensions, these shots aggregate into a clear probability distribution. In high dimensions, the same number of shots becomes spread so thinly that the results appear as a sparse, random scatter. This *Measurement Sparsity* means that the "audience's cheers" are no longer a coherent signal, but a chaotic sequence of individual claps. For the blind puppeteer, the cost landscape becomes "jittery" and unreliable, as the statistical noise of the sampling process effectively drowns out any surviving geometric signal.

In the interplay between the optimizer and Hilbert space, we are confronted not only with the noise of quantum hardware but also with the nature of the circuit's high-dimensional geometry.

6 Resolution: Navigating the Metric Mismatch

The fundamental friction in *Variational Quantum Algorithms* arises from a profound metric mismatch. Classical optimizers, such as Adam or BFGS, operate under the assumption of a flat, Euclidean landscape where the distance between two points is measured along a straight line. However, the *Parametrized Quantum Circuit* does not inhabit such a world. Instead, it moves within the *Fubini-Study metric* of the Hilbert space—a highly curved manifold where a small step in parameter space (θ) may result in a massive jump in the state space, or conversely, no meaningful change at all. To resolve this, we must provide the blind "classical-puppeteer" with a geometric compass, translating the optimizer's Euclidean intuition into the language of quantum geometry (Figure 7).

The resolution lies in the adoption of the "quantum-aware" optimization, specifically the *Quantum Natural Gradient* (QNG). By incorporating the *Fubini-Study metric* tensor directly into the update rule, we effectively even out the curved manifold from the optimizer's perspective. Crucially, this requires the classical cost function to account for the interplay between the manifold's curvature and the classical parameter space. We are not just looking at the shape of

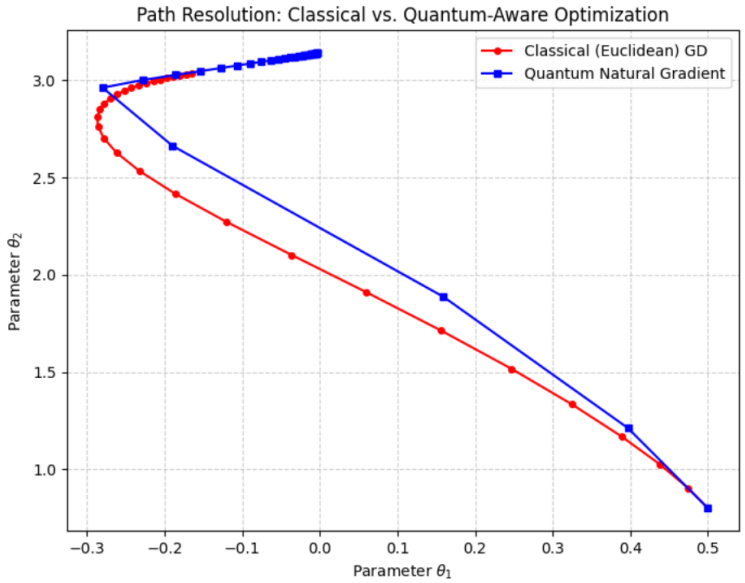


Figure 7: Euclidean vs. QNG optimization.

the Hilbert space in isolation but how our specific "classical-strings"—the PQC parameters—pull against that geometry. By scaling the gradient by the inverse of the metric, the puppeteer no longer pulls in the dark—every update is weighted by the local sensitivity of the state space relative to its parameters.

As an example, Figure 7 compares the efficiency of the classical and quantum-aware optimization of the circuit presented in Figure 2. Although the cost landscape in Figure 3b suggests a perilous path for any optimizer, Figure 7 (blue) demonstrates how geometric awareness fundamentally alters the optimization journey. By operating within the natural metric of the manifold shown in Figure 3a, the Quantum Natural Gradient effectively collapses the "warped" distances of the parameter space. It identifies the most direct route to the manifold's high-volume regions, bypassing the inefficient, curved trajectory that the classical gradient (red) is forced to follow due to its lack of geometric context.

The difference in final convergence (Figure 7, top-left), where the Quantum Natural Gradient (QNG) reaches a cost of ≈ -2.00 while the classical optimizer stalls at ≈ -1.992, illustrates the *Metric Mismatch*. Although this gap seems small in parameter space, it reflects a fundamental gap in the Hilbert space. In the vastness of the Hilbert space, the 0.008 difference in cost can represent a significant distance in state fidelity, marking the difference between a model that has truly captured the target phenomenon and one that is merely hovering near it. The classical "blind puppeteer" is clearly misled by the flattened Euclidean landscape, mistaking a shallow ridge for the minimum. In contrast, the QNG's geometric awareness reveals remaining room for improvement in state space, guiding the model to the true global minimum, before which the classical approach stalled.

However, geometry alone sometimes cannot escape the most resilient plateaus—regions so featureless that even a compass offers no heading. In these cases, we introduce stochastic perturbations, a deliberate "shaking" of the "puppet strings". By injecting controlled noise or using methods like SPSA, we break the symmetries that cause stagnation. These perturbations act as a kinetic energy boost, knocking the model off barren manifolds and pushing it toward

high-volume regions of Hilbert space where global optima are easier to find.

Ultimately, navigating the metric mismatch requires a synthesis of both sight and movement. Whether through the precise lens of the *Quantum Natural Gradient* or the strategic chaos of stochastic shaking, the goal remains the same: to harmonize the classical update with the quantum state. By reconciling the "classical-strings" with the Fubini-Study "quantum-stage," we transform a blind search into a directed exploration, ensuring that the optimizer's path respects the complex topography of the quantum landscape.

Much of the insights presented in this chapter were related to the paradoxes of the n-ball, where the volume is concentrated near the surface and points become orthonormal and equidistant, which are often dismissed as artifacts of purely random systems and not applicable to real-world QML modeling. In QML, however, apparent randomness arises from the *Metric Mismatch*. When a classical optimizer navigates a Hilbert-space manifold using a Euclidean cost landscape, the signal eventually vanishes, pushing the algorithm into stochastic drift. Then, its steps become statistically indistinguishable from random sampling, and the n-ball randomness requirements are met, trapping the model in the *Orthonormal Desert*. Only "quantum-aware" methods like QNG, which follows the manifold's geodesic curvature, can preserve the specificity needed to avoid this high-entropy extinction.

7 Conclusions: The Geometric Imperative

The journey through QML model development is ultimately a story of geometric translation. Success in training does not depend on the power of qubits alone, but on the alignment between the classical parameters and the quantum state space. By understanding the constraints of high-dimensional geometry—from the concentration of distance and measure, to the metric mismatch—quantum researchers and professionals can move beyond trial-and-error model development and begin to design architectures that are geometrically optimized for the quantum world.

This chapter explored the notion of the interplay between a classical optimizer and the Hilbert space, which identified the following key takeaways.

The Geometry of Measure as a Barrier: The *Curse of Dimensionality* is a physical transformation of the training landscape.

In business applications, such as those in logistics or finance, where we already face combinatorial "explosions," the n-ball concentration of volume ensures that the cost landscape is a featureless *Barren Plateau*. For these fields, training is not just a search—it is an effort to prevent the model's specialized business logic from dissolving into statistical noise.

The Metric Mismatch: Optimization fails when a classical optimizer's *Euclidean assumptions* clash with the *Fubini-Study metric* of the Hilbert space. Without accounting for the "warped" manifold in quantum space, classical optimization algorithms are essentially "flying blind" through non-linear quantum curvature.

The Transparency Gap and Sampling Barrier: The classical optimizer only perceives the quantum model through the lossy statistical filter of measurement. *Measurement sparsity* ensures that as the model scales, the resolution of our "instrument" drops, hiding the vanishing geometric signal beneath the noise floor.

False Convergence: Due to the possible presence of *Orthonormal Desert*, cost convergence in parameter space does not always ensure fidelity in Hilbert space. In a financial risk model, we

must beware "stagnation disguised as success", where a model appears stable during training but lacks the state-space precision to capture rare, high-impact tail risks.

The Necessity of the "Shake": Strategic, injected stochastic perturbations are a requirement, not an admission of hardware failure. Unlike passive hardware noise, which erodes the landscape, structured perturbations act as a geometric probe to break symmetries and force the optimizer out of stagnant, low-dimensional "potholes."

Quantum Natural Gradient: Classical optimizers navigate the cost landscape under the assumption of a flat parameter space, often resulting in a metric mismatch with the underlying quantum hardware. Conversely, the Quantum Natural Gradient (QNG) utilizes the Fubini-Study metric and the QFIM to perform optimization directly on the Hilbert space manifold. This ensures that the scale of parameter updates remains consistent with the actual change in the quantum state. Crucially, while QNG optimizes the path through state space, the optimality of the resulting candidates is still defined by a classical cost function, leaving the process tethered to the constraints of the classical task.

The "interplay" between a classical optimizer and the Hilbert space is a fundamental conflict of scale and perspective. We are attempting to use a linear Euclidean tool to navigate an exponential projective manifold. The difficulties discussed—*Barren Plateaus*, the *Orthonormal Desert*, and *Measurement Sparsity*—are not merely engineering hurdles to be overcome with more qubits or faster classical hardware. They are the natural consequences of high-dimensional geometry. To bridge this gap, we must move beyond the "black-box" optimization. By adopting a purely geometrical viewpoint—utilizing the *Fubini-Study metric* to correct our steps and injecting *strategic perturbations* to maintain our signal—we can begin to navigate the Hilbert space with "geometric honesty." The future of Quantum Machine Learning training lies not in fighting these geometric realities, but in building optimizers that actually speak the language of the manifold they are trying to master.

With these goals conquered, we will finally revel in the spectacle of the razor-eyed puppeteer, no longer groping in the dark, but commanding the curvature of the stage to whip his quantum marionettes into a passionate, purposeful dance.

References

[1] Cybulski, J. L. and Zając, S. (2025). Quantum Modelling of Time Series: Expressivity vs. Trainability.

[2] Lotan, S., Defienne, H., Talmon, R., and Bartal, G. (2025). Sparsity-Driven Entanglement Detection in High-Dimensional Quantum States. *arXiv preprint arXiv:2511.12546*.

[3] McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., and Neven, H. (2018). Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812.

[4] Schuld, M. and Petruccione, F. (2021). *Machine Learning with Quantum Computers*. Springer, 2nd ed. 2021 edition edition.

[5] Stokes, J., Izaac, J., Killoran, N., and Carleo, G. (2020). Quantum Natural Gradient. *Quantum*, 4:269.

From the Author

As a child in Warsaw, Poland, I grew up in a small two-bedroom apartment that, every Thursday, transformed into a bustling art studio. Among a dozen painters, poets, and sculptors, I spent my time drawing and dreaming of a life in the arts. Yet, I harbored a parallel passion for mathematics. In my specialist mathematical high school, I discovered that formal proofs were their own form of creation—requiring the same imagination and intuitive navigation through abstract inner spaces as painting.

When the time came to choose between fine arts, physics, and informatics, I saw the computer as the ultimate medium for expressing abstract ideas. This path led me to Australia, where I completed my studies in Computer Science and AI, eventually earning a PhD at the intersection of natural language processing and software engineering. After years in academia—ranging from information systems to business analytics—I found immersive 3D data visualization to be a perfect area for me to blend arts, mathematics, and computation into a single expression of truth in data and intuition in the eye of the beholder.

Seeking a field that would challenge me for decades to come, I turned toward quantum computing and quantum machine learning. It was a transition that demanded rigorous discipline and a new kind of experimentation. Today, I still consider myself an artist; only now, I paint my quantum models with the colors of state evolution across the high-dimensional manifolds of Hilbert space.