

Article

Not peer-reviewed version

Novel Wormholes in Computational Spacetime: Beyond Classical Algorithmic Shortcuts

[Michael Rey](#)*

Posted Date: 3 September 2025

doi: 10.20944/preprints202509.0313.v1

Keywords: computational spacetime; wormholes; algorithmic shortcuts; quantum computing; topological data analysis; distributed consensus; neuromorphic computing; differential privacy; federated learning



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Novel Wormholes in Computational Spacetime: Beyond Classical Algorithmic Shortcuts

Michael Rey 

Octonion Group, Hong Kong; contact@octoniongroup.com

Abstract

The geometric theory of computational spacetime models algorithmic efficiency as geodesics in a five-dimensional manifold (S, T, H, E, C) spanning space, time, memory hierarchy, energy, and quantum coherence. Classical “wormholes”—localized reparameterizations that bypass regions of high action—capture many known shortcuts (e.g., spectral sparsifiers, spanners, FFT/NTT, JL embeddings). This paper proposes seven new wormhole classes that leverage quantum–classical hybrids, topological structure, distributed consensus geometry, neuromorphic dynamics, differential privacy, and federated optimization. For each class we provide a formal setup, entry tolls, shortcut statements, structural preconditions, and resource trade-offs in the (S, T, H, E, C) framework. All complexity improvements are stated under explicit assumptions (e.g., QRAM-style access, partial synchrony, low-distortion embeddings, well-conditioned quantum Fisher information, sparsity, and idealized sketching/compression bounds). The analysis yields convergence conditions, approximation/stability bounds, and a comparative matrix to guide algorithm design in heterogeneous compute environments.

Keywords: computational spacetime; wormholes; algorithmic shortcuts; quantum computing; topological data analysis; distributed consensus; neuromorphic computing; differential privacy; federated learning

1. Introduction

Computational relativity frames algorithms as worldlines $\gamma : [0, 1] \rightarrow \mathcal{M}$ within a resource manifold $\mathcal{M} = (S, T, H, E, C)$ [1]. This geometric perspective reveals that algorithmic efficiency corresponds to finding shorter paths through computational spacetime, where each dimension represents a fundamental computational resource. Efficiency corresponds to minimizing an action functional; dramatic gains arise from “wormholes” that trade a localized entry toll for a shorter path through the manifold.

The concept of computational wormholes provides a unifying framework for understanding why certain algorithmic techniques achieve dramatic efficiency improvements. Just as wormholes in physics represent shortcuts through spacetime that connect distant regions, computational wormholes represent algorithmic techniques that allow computations to bypass expensive regions of the resource manifold by paying a modest upfront cost. This geometric interpretation helps explain the effectiveness of techniques like fast Fourier transforms, which trade $O(n \log n)$ preprocessing for exponential reductions in convolution complexity.

While classical examples are abundant—spanning geometric algorithms, algebraic techniques, and information-theoretic methods—emerging paradigms open new geometric avenues. Quantum-classical hybrid systems create new dimensions in computational spacetime where quantum coherence can be traded for classical computational resources. Topological data analysis reveals intrinsic geometric structure that can be exploited for compression. Distributed systems exhibit network geometries that can be leveraged for efficient consensus. Neuromorphic computing exploits temporal sparsity for energy efficiency. Privacy-preserving computation creates new constraint surfaces where novel optimization strategies may exist.

This work develops seven wormhole classes with rigorous but assumption-explicit guarantees. We (i) define wormholes via action reduction; (ii) formalize entry tolls ($\Delta S, \Delta T, \Delta H, \Delta E, \Delta C$); (iii) state shortcut bounds with clear preconditions; and (iv) compare resource trade-offs.

Contributions.

We formulate:

1. **Quantum-Topological Hybrid (QTH)** wormholes: These represent the first systematic approach to combining quantum amplitude estimation with topological data analysis. By encoding point clouds in quantum superposition states and using quantum algorithms to compute topological invariants, QTH wormholes suggest a quadratic lead-term improvement for persistent homology on quantum-accessible data. The key insight is that Betti numbers can be expressed as expectation values of quantum observables, enabling quantum speedups for their computation under QRAM-style access assumptions.
2. **Consensus-Preserving Distributed (CPD)** wormholes: These exploit the natural hyperbolic geometry of many real-world networks to achieve scalable Byzantine fault-tolerant consensus. By embedding network topology in hyperbolic space, CPD wormholes enable hierarchical consensus protocols that maintain safety and liveness guarantees while reducing communication complexity from quadratic to subquadratic. The negative curvature of hyperbolic space creates natural hierarchies that can be exploited for efficient message aggregation.
3. **Persistent Homology Compression (PHC)** wormholes: These leverage the stability theorem of persistent homology to create compressed representations that preserve essential topological features while dramatically reducing computational complexity. By selecting landmark points and constructing witness complexes, PHC wormholes yield $O(nk + k^3)$ pipelines that preserve salient topological signatures, enabling linear-time approximations to cubic-time exact computations.
4. **Variational Circuit Optimization (VCO)** wormholes: These address the notorious optimization challenges in variational quantum algorithms by exploiting the natural Riemannian geometry of quantum parameter manifolds. Using the quantum Fisher information metric to define natural gradients, VCO wormholes achieve linear-rate convergence under well-conditioned assumptions, potentially overcoming the barren plateau problem that plagues many quantum optimization landscapes.
5. **Neuromorphic Computing (NC)** wormholes: These exploit the event-driven nature of biological neural networks to achieve dramatic energy reductions for computations with sparse temporal structure. By encoding information in spike timing rather than continuous activation values, NC wormholes provide quadratic energy reductions for suitably sparse signals, enabling ultra-low-power computation for edge devices and IoT applications.
6. **Differential Privacy (DP)** wormholes: These navigate the complex three-way trade-off between privacy, utility, and computational efficiency. Through hierarchical noise injection combined with sketching techniques, DP wormholes offer pathways from quadratic to near-linear workloads while maintaining formal differential privacy guarantees, contingent on idealized sketching assumptions.
7. **Federated Learning (FL)** wormholes: These address the communication bottleneck in distributed machine learning through gradient compression and sketching techniques. FL wormholes reduce communication complexity versus dense baselines while preserving convergence guarantees under bounded-variance compression assumptions, enabling scalable federated optimization across resource-constrained devices.

2. Background: Computational Spacetime and Wormholes

The geometric theory of computational spacetime provides a mathematical framework for understanding algorithmic efficiency through the lens of differential geometry. This approach treats computational resources as coordinates in a Riemannian manifold, where algorithms correspond to trajectories and efficiency is measured by path length.

Let \mathcal{M} be a Riemannian manifold with metric $g_{\mu\nu}$ over coordinates (S, T, H, E, C) representing space complexity, time complexity, memory hierarchy costs, energy consumption, and quantum coherence respectively. The metric tensor encodes the relationships between different resources and reflects the characteristics of the computational platform. For example, on energy-constrained mobile devices, the energy components would be weighted more heavily than on high-performance computing clusters.

An algorithmic trajectory γ has action

$$\text{Action}(\gamma) = \int_0^1 \sqrt{g_{\mu\nu} \dot{\gamma}^\mu \dot{\gamma}^\nu} d\lambda, \quad (1)$$

with optimality characterized by geodesics subject to platform and problem constraints. This action functional captures the total “cost” of executing an algorithm, accounting for all resource dimensions simultaneously.

A *wormhole* is a transformation W for a task family \mathcal{F} and tolerance δ such that for each instance $x \in \mathcal{F}$,

$$\text{Action}(\gamma_x) - \left(\text{Action}(\text{enter } W) + \text{Action}(\gamma_{W(x)}^*) + \text{Action}(\text{exit } W) \right) \geq \Omega(\text{poly}(|x|)), \quad (2)$$

with correctness preserved up to δ . The vector $(\Delta S, \Delta T, \Delta H, \Delta E, \Delta C)$ is the entry toll, representing the upfront cost of constructing the wormhole.

This definition captures the essential trade-off in algorithmic shortcuts: by paying a localized entry toll, we can access a shorter path through computational spacetime that provides polynomial savings in the action integral. The tolerance parameter δ allows for approximate solutions, which is crucial for many practical applications where exact solutions are computationally intractable.

Representative classical wormholes.

The wormhole framework unifies many well-known algorithmic techniques. Spanners and hopsets add sparse edge sets to graphs, creating shortcuts for distance queries. Spectral sparsifiers preserve quadratic forms while reducing graph size, enabling near-linear algorithms for Laplacian systems. The Fast Fourier Transform converts convolution to pointwise multiplication, reducing complexity from quadratic to quasi-linear. Low-rank factorizations enable efficient updates to linear systems. Separator decompositions exploit graph structure for divide-and-conquer algorithms. Hub labeling creates distance oracles with sublinear query time. Johnson–Lindenstrauss embeddings preserve distances in lower dimensions. Proof-carrying computation allows verifiers to check results with minimal computational cost. All of these techniques instantiate the wormhole paradigm by trading preprocessing costs for dramatic runtime improvements [2–5].

3. Quantum–Topological Hybrid (QTH) Wormholes

Quantum-Topological Hybrid wormholes represent a novel fusion of quantum computing and topological data analysis, two fields that have traditionally developed independently. The key insight is that topological invariants like Betti numbers can be expressed as expectation values of quantum observables, potentially enabling quantum speedups for their computation.

3.1. Mathematical Framework

Topological data analysis seeks to understand the shape and structure of high-dimensional data by computing topological invariants across multiple scales. Given a point cloud $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$, classical persistent homology builds filtered simplicial complexes and computes their homology groups, typically requiring $O(n^3)$ time for the matrix reduction phase when dimension d is fixed.

The quantum approach begins with amplitude encoding, which represents the point cloud as a quantum superposition state. Under *QRAM-style* access assumptions, we can efficiently prepare the state

$$|\psi_X\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^n |i\rangle \otimes |x_i\rangle, \quad (3)$$

where each point is encoded in the amplitudes of a quantum state. This encoding enables quantum algorithms to process all points simultaneously through superposition.

A quantum oracle U_{complex} implements the characteristic function of simplices, marking which combinations of vertices form valid simplices in the filtered complex. This oracle encodes the geometric structure of the point cloud and enables quantum algorithms to reason about topological features.

The crucial observation is that Betti numbers, which count the number of topological holes in each dimension, can be expressed as expectation values:

$$\beta_k = \langle \psi_X | P_k | \psi_X \rangle, \quad P_k = \text{projector onto } k\text{th homology subspace}, \quad (4)$$

where P_k is the projector onto the k -th homology subspace. This projector can be constructed using quantum linear algebra techniques, specifically quantum singular value decomposition of the boundary matrices.

Quantum amplitude estimation then allows us to estimate β_k with precision ϵ using $\tilde{O}(1/\epsilon)$ queries to the quantum oracle, compared to the $O(1/\epsilon^2)$ samples required by classical Monte Carlo methods. This quadratic improvement in query complexity is the source of the potential quantum speedup.

3.2. Entry Toll Analysis

The construction of a QTH wormhole requires several preprocessing steps that constitute the entry toll:

State preparation: Encoding the point cloud into quantum amplitudes requires $O(n)$ quantum gates under QRAM assumptions. However, without QRAM, classical data loading can dominate the complexity, potentially negating the quantum advantage. This is a critical assumption that determines the viability of the approach.

Oracle construction: Building the quantum oracle for simplex detection depends on the geometric structure of the point cloud. For point clouds with bounded doubling dimension, efficient geometric predicates can be implemented with polynomial overhead. The oracle complexity scales with the intrinsic dimensionality of the data rather than the ambient dimension.

Circuit depth and coherence: The quantum circuits for homology computation have depth $\text{polylog}(n)$, which is favorable for near-term quantum devices. The coherence requirement is $\tilde{O}(\sqrt{k})$, driven by the amplitude estimation procedure, where k is the number of significant topological features.

3.3. Shortcut Guarantee (Assumption-Explicit)

For topologically sparse data satisfying $\sum_{j=0}^d \beta_j \leq \text{polylog}(n)$ and assuming efficient state preparation and oracle construction,

$$T_{\text{classical}} = O(n^3) \rightsquigarrow T_{\text{QTH}} = \tilde{O}(\sqrt{n}) \text{ queries to topological primitives.} \quad (5)$$

This represents a potential quadratic speedup in the leading term. However, the total runtime depends critically on the costs of data loading and oracle implementation. Without QRAM or efficient geometric oracles, the advantage may vanish entirely.

3.4. Structural Preconditions

QTH wormholes require several structural conditions to be effective:

Quantum accessibility: The point cloud must admit efficient quantum encoding, either through QRAM or specialized data structures that enable polynomial-overhead state preparation.

Geometric structure: The data should have bounded doubling dimension or similar geometric constraints that enable efficient simplex detection oracles.

Topological sparsity: The persistent homology should have bounded complexity, with the total number of significant topological features scaling at most polylogarithmically with the data size.

Coherence requirements: The quantum computation must be completed within the coherence time of the quantum device: $\tau_c \gtrsim \tilde{O}(\sqrt{n}) T_{\text{gate}}$, where T_{gate} is the elementary gate time.

3.5. Convergence, Approximation, Stability

The approximation quality of QTH wormholes depends on the precision of quantum amplitude estimation. Exact amplitude estimation gives precise Betti numbers, while approximate estimation introduces controlled errors. The stability of the approach follows from the stability theorem of persistent homology, which ensures that small perturbations in the input data lead to small changes in the persistence diagram, up to an additive error of $O(\epsilon)$ [6,7].

3.6. Resource Trade-offs

QTH wormholes exhibit the following resource trade-offs: $T \downarrow\downarrow$ (dramatic time reduction), $S \uparrow$ (increased space for quantum state storage), $H \downarrow$ (reduced memory hierarchy costs due to quantum parallelism), $E \sim$ (comparable energy consumption), $C \downarrow$ (quantum coherence consumption).

3.7. Applications

QTH wormholes enable several novel applications at the intersection of quantum computing and data analysis:

Quantum machine learning: Topological features extracted via QTH wormholes can serve as quantum-enhanced features for machine learning algorithms, potentially providing quantum advantages for classification and clustering tasks on high-dimensional data.

Quantum chemistry: Configuration spaces in quantum chemistry often have rich topological structure. QTH wormholes could enable efficient analysis of molecular conformations and reaction pathways, providing insights into chemical processes.

Quantum error correction: The topological structure of quantum error-correcting codes can be analyzed using QTH wormholes, potentially leading to improved code design and decoding algorithms.

4. Consensus-Preserving Distributed (CPD) Wormholes

Consensus-Preserving Distributed wormholes address one of the fundamental challenges in distributed systems: achieving agreement among nodes in the presence of Byzantine failures while maintaining communication efficiency. Classical Byzantine fault-tolerant protocols require quadratic message complexity, creating scalability bottlenecks for large networks.

4.1. Mathematical Framework

The key insight behind CPD wormholes is that many real-world networks exhibit hyperbolic geometry, where the negative curvature creates natural hierarchical structures that can be exploited for efficient consensus protocols.

Consider a distributed network modeled as a graph $G = (V, E)$ with $|V| = n$ nodes, where up to $f < n/3$ nodes may exhibit Byzantine behavior. Instead of treating the network as a flat Euclidean space, CPD wormholes embed the network topology in hyperbolic space \mathbb{H}^d with low distortion.

The hyperbolic distance between nodes u and v at coordinates $x_u, x_v \in \mathbb{H}^d$ is given by:

$$d_{\mathbb{H}}(u, v) = \operatorname{acosh}\left(1 + \frac{2\|x_u - x_v\|^2}{(1 - \|x_u\|^2)(1 - \|x_v\|^2)}\right). \quad (6)$$

This embedding creates natural neighborhoods $N_r(v)$ of size $\Theta(\log n)$ around each node, where nodes communicate primarily with geometrically nearby neighbors. The consensus protocol operates in three phases:

Local consensus: Nodes achieve consensus within their geometric neighborhoods using classical PBFT protocols. Since each neighborhood has logarithmic size, this phase requires $O(n \log^2 n)$ total messages.

Hierarchical aggregation: Local consensus results are aggregated through a hyperbolic tree structure that exploits the exponential growth of hyperbolic space. Each level of the tree reduces the number of active nodes by a constant factor while preserving Byzantine fault tolerance.

Global verification: A sparse set of $O(\sqrt{n})$ nodes performs global verification to ensure consistency across the hierarchical structure. The hyperbolic embedding ensures that this verification can detect Byzantine coalitions with high probability.

4.2. Entry Toll Analysis

The construction of a CPD wormhole requires several preprocessing steps:

Hyperbolic embedding: Computing the hyperbolic coordinates for all nodes requires solving an optimization problem to minimize embedding distortion while preserving network connectivity. This contributes $O(n^2)$ time complexity and $O(n \log n)$ space complexity for storing coordinates.

Cryptographic setup: Establishing cryptographic keys and authentication mechanisms for the hierarchical protocol requires $O(n^2)$ energy and time during initialization.

Neighborhood and hierarchy construction: Building the geometric neighborhoods and hierarchical tree structure requires $O(n \log n)$ memory hierarchy operations for efficient data structure construction.

4.3. Shortcut Guarantee

Under assumptions of partial synchrony and low-distortion hyperbolic embeddings:

Messages per round: $O(n^2) \rightsquigarrow O(n^{3/2})$ (generic); with stronger structure: $O(n \log n)$. (7)

The generic bound applies to arbitrary networks that admit hyperbolic embeddings, while the stronger bound applies to networks with particularly good hyperbolic structure, such as those arising from social networks, biological networks, or internet topologies.

4.4. Byzantine Fault Tolerance

CPD wormholes maintain Byzantine fault tolerance through geometric redundancy and probabilistic verification:

Local fault tolerance: Each geometric neighborhood can tolerate up to $\lfloor |N_r(v)|/3 \rfloor$ Byzantine nodes. Since neighborhoods are chosen to have limited overlap, a global Byzantine coalition of size f can control at most $O(f \log n)$ local neighborhoods.

Hierarchical resilience: The hyperbolic tree structure ensures that Byzantine nodes cannot control critical paths in the aggregation phase. Any Byzantine coalition of size $f < n/3$ can influence at most $O(\log n)$ tree nodes at each level.

Global verification: The sparse verification phase uses random sampling to detect inconsistencies with high probability. The hyperbolic geometry ensures that Byzantine coalitions cannot hide from this verification process.

Safety and liveness guarantees derive from the combination of local BFT protocols and geometric redundancy, following the analysis of classical Byzantine consensus protocols [8].

4.5. Structural Preconditions & Convergence

CPD wormholes require several structural conditions:

Partial synchrony: The network must satisfy partial synchrony assumptions, where message delays are bounded but the bound may be unknown.

Hyperbolic embeddability: The network topology must admit a low-distortion embedding in hyperbolic space. This condition is satisfied by many real-world networks that exhibit hierarchical structure.

Bounded degree: Nodes should have bounded degree to ensure that local neighborhoods have manageable size.

The hierarchical structure with $O(\log n)$ levels yields $O(\log n)$ consensus rounds overall, providing logarithmic latency for global agreement.

4.6. Resource Trade-offs & Applications

CPD wormholes exhibit the following resource trade-offs: $H \downarrow$ (reduced memory hierarchy costs through locality), $E \downarrow$ (reduced energy consumption due to fewer messages), $S \uparrow$ (increased space for storing coordinates and tree structure).

Applications include:

Blockchain consensus: CPD wormholes can improve the scalability of blockchain consensus protocols by reducing the communication overhead while maintaining security guarantees.

Federated aggregation: In federated learning systems, CPD wormholes can enable efficient aggregation of model updates while providing robustness against malicious participants.

IoT and edge networks: Resource-constrained IoT devices can benefit from the reduced communication requirements of CPD wormholes while maintaining fault tolerance.

5. Persistent Homology Compression (PHC) Wormholes

Persistent Homology Compression wormholes address the computational bottleneck in topological data analysis by exploiting the stability of topological invariants. The key insight is that many datasets have intrinsic topological structure that can be captured by a much smaller representative subset, enabling dramatic computational savings while preserving essential geometric information.

5.1. Mathematical Framework

The approach is based on landmark selection and witness complex construction, which provides a principled way to compress high-dimensional datasets while preserving their topological signatures.

Given a point cloud $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$, we select $k \ll n$ landmark points using farthest-point sampling:

$$L = \{l_1, \dots, l_k\}, \quad l_{i+1} = \arg \max_{x \in X} \min_{j \leq i} d(x, l_j). \quad (8)$$

This greedy algorithm ensures that landmarks are well-distributed throughout the point cloud, providing good coverage of the underlying geometric structure. The farthest-point sampling strategy is particularly effective for datasets with intrinsic low-dimensional structure.

We then construct a witness complex $W(L, X)$ where the full point cloud X serves as witnesses for simplices formed by landmarks. A simplex $\sigma = \{l_{i_0}, \dots, l_{i_p}\}$ is admitted to the complex if there exists a witness point $w \in X$ such that:

$$d(w, l_{i_j}) \leq d(w, l) + \nu \quad \forall l \notin \sigma, \forall j, \quad (9)$$

where $\nu \geq 0$ is a relaxation parameter that controls the strictness of the witness condition.

This construction ensures that the witness complex captures the topological features that are supported by the data, while filtering out spurious features that arise from sampling artifacts or noise.

5.2. Entry Toll Analysis

The construction of a PHC wormhole involves several computational steps:

Landmark selection: Computing the farthest-point sampling requires $O(nk)$ distance computations, where each new landmark requires finding the point farthest from all previously selected landmarks.

Witness complex construction: Building the witness complex requires $O(k^{d+1})$ time to enumerate potential simplices and check witness conditions. The complexity depends on the intrinsic dimension rather than the ambient dimension.

Compressed boundary matrices: The resulting simplicial complex has $O(k^{d+1})$ simplices, leading to boundary matrices of size $O(k^{d+1})$ rather than the $O(n^{d+1})$ size of the full complex.

5.3. Shortcut Guarantee

For datasets with intrinsic low-dimensional structure and topological sparsity:

$$T_{\text{classical}} = O(n^3) \rightsquigarrow T_{\text{PHC}} = O(nk) + O(k^{d+1}) + O(k^3). \quad (10)$$

When $k \ll n$ (often $k = \text{polylog}(n)$ for datasets with good structure), this represents a substantial reduction in computational complexity. The savings are particularly dramatic for large datasets where the full persistent homology computation would be intractable.

5.4. Approximation Guarantees & Stability

The theoretical foundation for PHC wormholes rests on the stability theorem of persistent homology, which provides approximation guarantees for the compressed representation.

The bottleneck stability theorem ensures that:

$$d_B(\text{PD}(X), \text{PD}(S_k(X))) \leq 2\varepsilon, \quad (11)$$

where d_B is the bottleneck distance between persistence diagrams, $\text{PD}(X)$ is the persistence diagram of the full dataset, $\text{PD}(S_k(X))$ is the persistence diagram of the compressed representation, and ε is the approximation error.

Furthermore, topological features with persistence greater than 2ε are preserved within ε error in both birth and death times. This means that the most significant topological features are accurately captured by the compressed representation, while only the least persistent (and typically least important) features may be lost or distorted.

5.5. Structural Preconditions & Convergence

PHC wormholes are most effective under several structural conditions:

Low intrinsic dimension: The dataset should have intrinsic dimension much smaller than the ambient dimension, enabling effective compression through landmark selection.

Topological sparsity: The persistent homology should have a clear separation between significant and insignificant features, with most topological information concentrated in a small number of highly persistent features.

Clear persistence gap: There should be a clear gap in the persistence values, allowing for natural thresholding to separate signal from noise.

The approximation error decreases as $\varepsilon(k) = O((\text{vol}(X)/k)^{1/d})$, where d is the intrinsic dimension. This provides guidance for choosing the number of landmarks based on the desired approximation quality. The approach is stable under perturbations, inheriting the stability properties of persistent homology [3,6,7].

5.6. Resource Trade-offs & Applications

PHC wormholes exhibit favorable resource trade-offs: $T \downarrow\downarrow$ (dramatic time reduction), $S \downarrow$ (reduced space complexity), $H \downarrow$ (improved memory hierarchy performance due to smaller working sets).

Applications span multiple domains:

Large-scale shape analysis: PHC wormholes enable topological analysis of massive 3D shape collections, such as those arising in computer graphics, medical imaging, and materials science.

Time-series topological data analysis: For time-varying datasets, PHC wormholes can track the evolution of topological features efficiently, enabling real-time monitoring of dynamic systems.

Machine learning features: Compressed topological signatures can serve as robust features for machine learning algorithms, providing geometric insights that complement traditional statistical features.

6. Variational Circuit Optimization (VCO) Wormholes

Variational Circuit Optimization wormholes address one of the most challenging problems in quantum computing: optimizing the parameters of variational quantum algorithms. These algorithms are central to near-term quantum computing applications but suffer from notoriously difficult optimization landscapes characterized by barren plateaus, local minima, and exponentially small gradients.

6.1. Mathematical Framework

The fundamental challenge in variational quantum algorithms is optimizing a parameterized quantum circuit $U(\theta)$ with parameters $\theta \in \mathbb{R}^p$ to minimize an objective function. The quantum state prepared by the circuit is $|\psi(\theta)\rangle = U(\theta)|0\rangle$, and the optimization objective is typically the expectation value of a Hamiltonian:

$$L(\theta) = \langle \psi(\theta) | H | \psi(\theta) \rangle. \quad (12)$$

Classical optimization methods treat the parameter space as Euclidean, using standard gradient descent or more sophisticated techniques like ADAM or L-BFGS. However, this approach ignores the intrinsic geometry of the quantum state manifold, which can lead to inefficient optimization trajectories and poor convergence properties.

VCO wormholes exploit the natural Riemannian structure of the quantum state manifold by using the Quantum Fisher Information Matrix (QFIM) as the metric tensor:

$$F_{ij} = 4 \Re[\langle \partial_i \psi | \partial_j \psi \rangle - \langle \partial_i \psi | \psi \rangle \langle \psi | \partial_j \psi \rangle], \quad (13)$$

where $|\partial_i \psi\rangle = \partial_{\theta_i} |\psi(\theta)\rangle$.

The QFIM encodes the sensitivity of the quantum state to parameter changes and provides the natural metric for measuring distances in parameter space. This leads to the natural gradient update rule:

$$\theta_{t+1} = \theta_t - \eta F^{-1} \nabla L, \quad (14)$$

which follows geodesics in the Riemannian manifold defined by the QFIM rather than straight lines in Euclidean parameter space.

6.2. Entry Toll Analysis

The implementation of VCO wormholes requires several computational steps that constitute the entry toll:

QFIM estimation: Computing the QFIM requires $O(p^2)$ shifted-circuit evaluations, where each matrix element is estimated using the parameter-shift rule or finite differences. This represents a significant overhead compared to standard gradient computation.

Matrix inversion: Inverting the QFIM requires $O(p^3)$ classical computation, though this can be reduced using iterative methods or low-rank approximations when the QFIM has special structure.

Gradient computation: Computing the objective function gradients requires $O(p)$ circuit evaluations per optimization step, which is comparable to classical methods.

6.3. Shortcut Guarantee & Convergence

The power of VCO wormholes lies in their superior convergence properties. When the QFIM is well-conditioned, satisfying $\lambda_{\min}I \preceq F \preceq \lambda_{\max}I$ for positive constants λ_{\min} and λ_{\max} , the natural gradient method achieves linear convergence:

$$L(\theta_t) - L^* \leq \left(1 - \eta \frac{\lambda_{\min}}{\lambda_{\max}}\right)^t (L(\theta_0) - L^*), \quad (15)$$

with optimal step size $\eta^* = \frac{2}{\lambda_{\min} + \lambda_{\max}}$.

This represents a dramatic improvement over standard gradient descent, which typically achieves only sublinear convergence rates on quantum optimization landscapes. The geometric insight is that natural gradients automatically adapt to the local curvature of the quantum state manifold, taking larger steps in directions where the state changes slowly and smaller steps where it changes rapidly [9,10].

6.4. Structural Preconditions & Practicalities

VCO wormholes are most effective under several conditions:

Well-conditioned QFIM: The quantum Fisher information matrix should be well-conditioned, which often occurs for local circuit architectures where parameters affect the quantum state in a balanced way.

Efficient QFIM computation: For practical implementation, the QFIM should admit efficient computation through techniques like block-diagonal approximations, low-rank factorizations, or stochastic estimation.

Adaptive regularization: In practice, the QFIM may become ill-conditioned, requiring regularization techniques like $F + \lambda I$ to ensure numerical stability.

While VCO wormholes can mitigate some aspects of the barren plateau problem through improved geometry, they do not eliminate it entirely. The fundamental issue of exponentially small gradients in deep quantum circuits remains a challenge that requires complementary techniques like parameter initialization strategies and circuit architecture design.

6.5. Resource Trade-offs & Applications

VCO wormholes exhibit the following trade-offs: $T \downarrow$ in terms of optimization iterations, but per-iteration cost increases with $C \uparrow$ due to additional circuit evaluations for QFIM estimation.

Key applications include:

Quantum machine learning: VCO wormholes can improve the training of quantum neural networks and quantum kernel methods by providing more efficient optimization trajectories.

Quantum Approximate Optimization Algorithm (QAOA): For combinatorial optimization problems, VCO wormholes can help find better parameter settings more quickly, potentially improving the approximation ratios achieved by QAOA.

Variational Quantum Eigensolver (VQE): In quantum chemistry applications, VCO wormholes can accelerate the search for ground state energies and molecular properties.

Quantum control: For quantum control problems, VCO wormholes provide a principled approach to optimizing control pulses while respecting the geometric constraints of quantum dynamics.

7. Neuromorphic Computing (NC) Wormholes

Neuromorphic Computing wormholes represent a paradigm shift from traditional digital computation to event-driven processing inspired by biological neural networks. These wormholes exploit the temporal sparsity inherent in many real-world signals to achieve dramatic energy reductions while maintaining computational accuracy.

7.1. Mathematical Framework

The foundation of neuromorphic computing lies in the event-driven nature of biological neural networks, where information is encoded in the timing of discrete spike events rather than continuous analog values. The basic computational unit is the Leaky Integrate-and-Fire (LIF) neuron model:

$$\tau_m \dot{V}_i = -V_i + R \sum_j w_{ij} I_j(t), \quad (16)$$

where $V_i(t)$ is the membrane potential of neuron i , τ_m is the membrane time constant, R is the membrane resistance, w_{ij} are synaptic weights, and $I_j(t)$ represents input currents from presynaptic neurons.

The key innovation is that input currents are represented as sequences of discrete spike events:

$$I_j(t) = \sum_k \alpha_k \delta(t - t_k^{(j)}), \quad (17)$$

where $t_k^{(j)}$ are the spike times for neuron j and α_k are the spike amplitudes.

This event-driven representation enables energy consumption that is proportional to activity rather than computation time:

$$E_{\text{total}} = \sum_{i,k} E_{\text{spike}} \delta(t - t_k^{(i)}) + \sum_{i,j,k} E_{\text{syn}} w_{ij} \delta(t - t_k^{(j)}). \quad (18)$$

The first term represents the energy cost of generating spikes, while the second term represents the energy cost of synaptic transmission. Both are event-driven, meaning that energy is consumed only when spikes occur, not during periods of inactivity.

7.2. Entry Toll & Shortcut Guarantee

The construction of NC wormholes requires several preprocessing steps:

Spike encoding: Converting input signals to spike trains requires temporal encoding schemes, contributing $O(n)$ preprocessing time where n is the input dimension.

Sparse connectivity initialization: Setting up the synaptic connectivity matrix with sparse topologies requires careful initialization to balance connectivity and efficiency.

For signals with temporal sparsity $\rho \ll 1$ and maximum firing rate f_{max} , NC wormholes achieve:

$$E_{\text{classical}} = \Theta(n^2 CV^2) \rightsquigarrow E_{\text{neuro}} = \tilde{O}(\rho n) \cdot (CV^2/\alpha), \quad (19)$$

where $\alpha \gg 1$ is the neuromorphic efficiency factor. This supports quadratic energy reductions in sparse regimes, making NC wormholes particularly attractive for battery-powered and edge computing applications [11].

7.3. Temporal Coding, STDP, Structure

NC wormholes exploit several biological mechanisms for efficient computation:

Temporal coding: Information is encoded in the precise timing of spikes rather than just their rates, enabling rich representational capabilities with sparse spike trains.

Rate coding: Information can also be encoded in average firing rates over time windows, providing a more robust but less efficient encoding scheme.

Population coding: Information is distributed across populations of neurons, providing redundancy and fault tolerance.

Spike-Timing-Dependent Plasticity (STDP): Synaptic weights adapt based on the relative timing of pre- and post-synaptic spikes:

$$\Delta w_{ij} = \begin{cases} A_+ e^{-\Delta t / \tau_+} & \text{if } \Delta t > 0 \\ -A_- e^{\Delta t / \tau_-} & \text{if } \Delta t < 0 \end{cases} \quad (20)$$

where $\Delta t = t_j - t_i$ is the spike time difference. This enables unsupervised learning and adaptation to input statistics.

The event-driven nature provides inherent robustness to noise and variability, making NC wormholes suitable for real-world applications where perfect precision is not required.

7.4. Resource Trade-offs & Applications

NC wormholes exhibit dramatic energy reductions ($E \downarrow$) while maintaining comparable performance in other dimensions. There are trade-offs between energy efficiency and computational precision, but many applications can tolerate the reduced precision in exchange for massive energy savings.

Applications are particularly compelling for resource-constrained environments:

Edge sensing and processing: NC wormholes enable ultra-low-power processing of sensory data in IoT devices, extending battery life by orders of magnitude.

Robotics and autonomous systems: Real-time control and navigation systems can benefit from the low latency and energy efficiency of neuromorphic processing.

Brain-computer interfaces: The biological compatibility of neuromorphic processing makes it ideal for interfacing with neural signals and prosthetic devices.

8. Differential Privacy (DP) Wormholes

Differential Privacy wormholes address the fundamental tension between privacy protection, computational efficiency, and utility preservation in data analysis. These wormholes navigate the complex three-way trade-off space by exploiting geometric structure in privacy-preserving computation.

8.1. Mathematical Framework

Differential privacy provides formal guarantees about the privacy protection offered by randomized algorithms. For a dataset D and privacy parameters (ϵ, δ) , the exponential mechanism provides a principled way to select outputs while preserving privacy:

$$P(r) \propto \exp\left(\frac{\epsilon q(D, r)}{2\Delta q}\right), \quad (21)$$

where $q(D, r)$ is a quality function that measures how good output r is for dataset D , and Δq is the global sensitivity of the quality function.

DP wormholes combine this privacy mechanism with hierarchical aggregation and sketching techniques to reduce computational complexity while maintaining privacy guarantees. The key insight is that many data analysis tasks can be decomposed into hierarchical structures where privacy noise can be added at multiple levels, and sketching techniques can reduce the dimensionality of the computation.

8.2. Entry Toll & Shortcut Statement

The construction of DP wormholes requires several preprocessing steps:

Privacy budget allocation: Carefully dividing the privacy budget ϵ across different components of the computation requires $O(\log(1/\delta))$ space for accounting.

Noise calibration: Computing the appropriate noise levels for different sensitivity values requires careful analysis of the algorithm structure.

Under *idealized sketching guarantees* and hierarchical aggregation assumptions:

$$T_{\text{classical}} = \Theta(n^2) \rightsquigarrow T_{\text{DP-sketch}} = \tilde{O}\left(\frac{n \log n}{\epsilon}\right). \quad (22)$$

This represents a programmatic pathway from quadratic to near-linear complexity. However, the viability depends critically on the sketch error bounds and how the privacy budget is allocated across different levels of the hierarchy. The dependence on $1/\epsilon$ reflects the fundamental trade-off between privacy and computational efficiency [12].

8.3. Structural Preconditions, Trade-offs, Applications

DP wormholes are most effective under several conditions:

Data sparsity and sketchability: The underlying data should admit efficient sketching representations that preserve the essential statistical properties needed for the analysis task.

Acceptable privacy-utility regime: The application should be able to tolerate the noise introduced by the privacy mechanism while still obtaining useful results.

Hierarchical structure: The computation should admit natural hierarchical decomposition that enables efficient privacy budget allocation.

Resource trade-offs include increased space complexity ($S \uparrow$) for storing sketches and privacy accounting information, while potentially achieving significant time complexity reductions.

Applications span multiple domains where privacy is critical:

Private analytics: Large-scale data analysis tasks like computing statistics, training machine learning models, or performing exploratory data analysis while protecting individual privacy.

Secure telemetry: Collecting and analyzing system performance data, user behavior analytics, or sensor readings while preserving privacy.

Federated analytics: Aggregating insights across multiple organizations or devices while ensuring that sensitive information is not leaked.

9. Federated Learning (FL) Wormholes

Federated Learning wormholes address the communication bottleneck that limits the scalability of distributed machine learning. In federated settings, the cost of communicating model updates often dominates the total training time, especially when clients have limited bandwidth or intermittent connectivity.

9.1. Mathematical Framework

The core challenge in federated learning is that clients must communicate their local model updates to a central server for aggregation. In the standard approach, each client sends its full gradient vector, requiring $\Theta(d)$ communication per client per round, where d is the model dimension.

FL wormholes use compression techniques to reduce this communication burden. Clients send compressed gradients $g_i \mapsto \tilde{g}_i = C(g_i)$ where C is a compression operator with bounded variance:

$$\mathbb{E}\|\tilde{g}_i - g_i\|^2 \leq \sigma^2. \quad (23)$$

The compression can take many forms: quantization, sparsification, low-rank approximation, or sketching. The key requirement is that the compression error is bounded, ensuring that the optimization algorithm can still converge to the optimal solution.

9.2. Entry Toll & Shortcut Statement

The implementation of FL wormholes requires:

Compression and sketching infrastructure: Each client needs $O(d \log d)$ memory to store compression data structures and $O(d)$ time per round to perform compression.

Error correction mechanisms: To handle compression errors, additional mechanisms like error feedback or variance reduction may be needed.

In favorable compression regimes:

$$\text{Comm}_{\text{dense}} = \Theta(Td) \rightsquigarrow \text{Comm}_{\text{compressed}} = \tilde{O}(T\sqrt{d}), \quad (24)$$

where T is the number of communication rounds. This represents a significant reduction in communication complexity, particularly for high-dimensional models where d can be millions or billions of parameters.

The convergence analysis shows that under bounded compression variance and appropriate stepsize control, the compressed algorithm converges to the same solution as the uncompressed version, with convergence rate depending on the compression quality [13].

9.3. Structural Preconditions, Trade-offs, Applications

FL wormholes are most effective under several conditions:

Compression unbiasedness or bounded variance: The compression operator should either be unbiased ($\mathbb{E}[\tilde{g}_i] = g_i$) or have bounded variance to ensure convergence.

Heterogeneity handling: The system should be able to handle statistical and system heterogeneity across clients, which can affect the effectiveness of compression techniques.

Communication constraints: The benefits are most pronounced when communication is the bottleneck rather than local computation.

Resource trade-offs include increased local computation and memory ($S \uparrow$) for compression, reduced memory hierarchy costs ($H \downarrow$) due to smaller message sizes, and dramatically reduced communication costs.

Applications are particularly important in resource-constrained distributed settings:

On-device machine learning: Training models across mobile devices, IoT sensors, or edge computing nodes where bandwidth is limited and expensive.

Cross-silo federated learning: Collaboration between organizations (hospitals, banks, etc.) where data cannot be shared directly but models can be trained jointly.

Satellite and remote sensing: Training models on data collected by satellites or remote sensors where communication links have limited bandwidth and high latency.

10. Comparative Analysis

The seven novel wormhole classes exhibit distinct characteristics and trade-offs within the computational spacetime framework. Understanding these relationships is crucial for selecting the appropriate wormhole type for specific applications and computational environments.

Table 1. Resource trade-offs in (S, T, H, E, C). These proposed bounds hold *only* under the following assumptions: QRAM-style access (QTH), partial synchrony and low-distortion hyperbolic embeddings (CPD), topological sparsity (PHC), well-conditioned quantum Fisher information matrix (VCO), temporal sparsity (NC), and idealized sketching/compression bounds (DP/FL). Arrows: \uparrow increased cost, \downarrow reduced cost, \sim comparable, $-$ not applicable.

Wormhole	Classical \rightsquigarrow Proposed	S	H	E	C
QTH	$O(n^3) \rightsquigarrow \tilde{O}(\sqrt{n})$	\uparrow	\downarrow	\sim	\downarrow
CPD	$O(n^2) \rightsquigarrow O(n^{3/2}) \sim O(n \log n)$	\uparrow	\downarrow	\downarrow	$-$
PHC	$O(n^3) \rightsquigarrow O(nk + k^3)$	\downarrow	\downarrow	\sim	$-$
VCO	$O(\epsilon^{-2}) \rightsquigarrow O(\log(1/\epsilon))$	\uparrow	\sim	\sim	\uparrow
NC	$O(n^2) \rightsquigarrow \tilde{O}(\rho n)$	\sim	\sim	$\downarrow\downarrow$	$-$
DP	$O(n^2) \rightsquigarrow \tilde{O}(n \log n / \epsilon)$	\uparrow	\sim	\sim	$-$
FL	$Td \rightsquigarrow \tilde{O}(T\sqrt{d})$	\uparrow	\downarrow	\sim	$-$

10.1. Geometric Relationships and Synergies

The wormhole classes exhibit several interesting geometric relationships that suggest potential synergies:

Quantum-classical duality: QTH and VCO wormholes both exploit quantum geometry but in complementary ways. QTH uses quantum parallelism for classical topological problems, while VCO uses classical optimization theory for quantum parameter spaces. These could potentially be combined for quantum topological optimization.

Hierarchical structure exploitation: CPD, PHC, and FL wormholes all leverage hierarchical decomposition but in different domains. This suggests that hybrid approaches combining network topology, topological features, and model parameters could yield even greater efficiency gains.

Sparsity as a unifying theme: NC, PHC, and DP wormholes all benefit from different types of sparsity (temporal, topological, and data sparsity respectively). Understanding the relationships between these sparsity types could lead to unified compression techniques.

Trade-off complementarity: Different wormholes optimize different resource dimensions, suggesting that combinations could achieve better overall efficiency. For example, NC wormholes excel at energy reduction while PHC wormholes excel at time complexity reduction.

10.2. Selection Criteria and Application Domains

The choice of wormhole type depends on several factors:

Resource constraints: Energy-constrained environments favor NC wormholes, while time-critical applications may prefer QTH or PHC wormholes.

Data characteristics: High-dimensional data with topological structure benefits from PHC wormholes, while sparse temporal data is ideal for NC wormholes.

System architecture: Distributed systems can leverage CPD and FL wormholes, while quantum-classical hybrid systems can exploit QTH and VCO wormholes.

Privacy requirements: Applications requiring formal privacy guarantees should consider DP wormholes, potentially in combination with FL wormholes for distributed private learning.

11. Conclusions and Future Directions

This work has introduced seven novel classes of computational wormholes that exploit emerging paradigms in quantum computing, topological data analysis, distributed systems, neuromorphic computing, and privacy-preserving computation. Each wormhole class provides a rigorous mathematical framework with explicit assumptions, entry toll analysis, shortcut guarantees, and resource trade-offs within the computational spacetime formalism.

The key insight underlying all these wormholes is that modern computational challenges require algorithmic approaches that can navigate complex trade-offs between multiple resource dimensions simultaneously. Traditional algorithm design, which optimizes individual resources in isolation, may not capture the geometric relationships that enable significant efficiency improvements through coordinated resource management.

Theoretical contributions: We have established formal frameworks for understanding how quantum-classical interfaces, topological structure, network geometry, temporal sparsity, privacy constraints, and communication bottlenecks create new opportunities for algorithmic shortcuts. Each framework includes convergence analysis, approximation bounds, and stability conditions that ensure the wormholes maintain essential algorithmic properties.

Practical implications: The wormhole classes provide conceptual tools for algorithm design in heterogeneous computing environments. They suggest new research directions at the intersections of quantum computing and topology, distributed systems and geometry, neuromorphic computing and sparsity, and privacy and efficiency.

Future research directions: Several important questions remain open. How can multiple wormhole types be combined to achieve even greater efficiency gains? Can adaptive wormhole selection

strategies automatically choose the best approach based on problem characteristics? How do these concepts extend to emerging paradigms like optical computing, DNA computing, or quantum-classical hybrid systems?

The geometric perspective provided by computational spacetime theory offers a unifying framework for understanding these diverse algorithmic techniques. As computing systems become increasingly heterogeneous and resource-constrained, the ability to navigate the complex trade-offs between time, space, energy, and other resources will become ever more critical. The wormhole classes developed in this work provide a foundation for this navigation, offering both theoretical insights and practical algorithmic tools for next-generation computing systems.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflicts of interest.

AI and Machine Learning Statement: This work was developed with assistance from AI tools for literature review, mathematical derivation verification, and manuscript preparation. All novel theoretical contributions, mathematical frameworks, and algorithmic insights represent original research by the author. The AI assistance was used primarily for formatting, reference management, and ensuring mathematical notation consistency.

References

1. M. Rey, "Computational Relativity: A Geometric Theory of Algorithmic Spacetime," *Preprints*, 2025. <https://doi.org/10.20944/preprints202509.0015.v1>
2. H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction*, American Mathematical Society, 2010.
3. A. Zomorodian and G. Carlsson, "Computing persistent homology," *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.
4. J. E. Hopcroft, W. J. Paul, and L. G. Valiant, "On time versus space," *Journal of the ACM*, vol. 24, no. 2, pp. 332–337, 1977.
5. V. V. Williams, "On some fine-grained questions in algorithms and complexity," *Proceedings of the International Congress of Mathematicians*, pp. 3447–3487, 2018.
6. N. Otter et al., "A roadmap for the computation of persistent homology," *EPJ Data Science*, vol. 6, no. 1, pp. 1–38, 2017.
7. F. Chazal and B. Michel, "An introduction to topological data analysis: fundamental and practical aspects for data scientists," *Frontiers in Artificial Intelligence*, vol. 4, p. 667963, 2021.
8. M. Castro and B. Liskov, "Practical Byzantine fault tolerance," *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, pp. 173–186, 1999.
9. S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.
10. M. Cerezo et al., "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.
11. W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
12. C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Theory of Cryptography Conference*, pp. 265–284, 2006.
13. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.