Theory of Curved Computation Background-Independent Universal Law

Section 1: Introduction: From Fixed Logics to Emergent Geometries

The history of science is marked by paradigm shifts that replace static, absolute frameworks with dynamic, relational ones. The transition from Newtonian mechanics to Einstein's General Theory of Relativity stands as the paramount example, where the fixed stage of absolute space and time was supplanted by a dynamic spacetime manifold whose geometry is shaped by its matter-energy content. In the domain of computation, a similar revolution is overdue. The prevailing theoretical models, from the Turing machine to the von Neumann architecture, are fundamentally Newtonian in their conception. They operate on a fixed background—a static tape, a pre-defined memory space, an immutable instruction set—upon which the drama of computation unfolds. This report introduces a new paradigm, the Universal Law of Curved Computation, that recasts computation in Einsteinian terms. It posits that the space of possible computations is not a fixed stage but a dynamic manifold whose geometry is determined by the structure of the information it contains. In this view, computation is not the execution of externally imposed rules but an emergent process of geodesic motion through a curved informational landscape.

1.1 The Newtonian View of Computation

Traditional models of computation are inherently background-dependent. A background-dependent theory is one that possesses fixed, non-dynamical structures that are put in place "by hand" rather than emerging from the theory's own equations. The Turing machine, with its infinite tape and finite state machine, operates within a pre-defined, absolute framework. The rules of transition are fixed, and the geometry of the computational space—a one-dimensional discrete line—is unchanging. Similarly, the von Neumann architecture, which underpins nearly all modern computing, is built upon the background of a fixed, addressable memory and a central processing unit with a static instruction set. The logic is imposed upon the hardware, and the "laws of computation" are extrinsic to the data being processed.
This structure mirrors the classical physics of Newton, where space and time form an unchangeable, absolute backdrop against which the laws of motion play out. The geometry of this Newtonian world is Euclidean and fixed; it influences the motion of objects but is never influenced by them. In the same way, the "computational space" of a conventional algorithm is a fixed data structure (an array, a graph) whose properties are defined a priori, and the algorithm's logic dictates movement within this space without ever altering its fundamental geometric character. While powerful, this Newtonian view relegates computation to the realm of abstract symbol manipulation on a passive substrate, failing to capture the dynamic, self-organizing nature of information processing observed in complex physical and biological systems.

1.2 The Einsteinian Leap: Geometry as a Dynamic Field

General Relativity (GR) initiated a profound conceptual shift by unifying space, time, and gravitation into a single dynamic entity: spacetime. The central insight of GR is that the geometry of spacetime is not fixed but is a dynamical field that both acts upon and is acted upon by the distribution of mass and energy within it. The famous dictum of John Archibald Wheeler encapsulates this feedback loop: "Spacetime tells matter how to move; matter tells spacetime how to curve." This principle of a dynamic, relational geometry is the essence of background independence.

The Universal Law of Curved Computation proposes an analogous leap for the theory of information. It posits that the space of possible computational states is not a pre-defined, static structure but a dynamic manifold whose geometry is actively shaped by its informational content. This moves the theory from a simple restatement of equations to a predictive framework. Just as GR revealed that the presence of mass-energy leads to observable phenomena like gravitational lensing and time dilation, this new paradigm suggests that the structure of information should give rise to analogous computational effects. A region of high information density might act as a "computational lens," deflecting execution paths toward it. Traversing a region of high causal complexity could lead to "computational time dilation," where the number of state transitions required to cross it increases relative to simpler regions. The most extreme concentrations of informational structure might even form "computational singularities"—non-halting states or informational black holes from which no geodesic can escape. The analogy is not merely metaphorical; it suggests a deep structural isomorphism where information is to computation what mass-energy is to spacetime.

This perspective fundamentally redefines the relationship between software and hardware. In the conventional view, software is an abstract set of instructions executed on a fixed hardware substrate. In the proposed framework, the distinction blurs. The "software"—the information content, its probabilistic nature, and its causal structure—dynamically defines the effective "hardware"—the geometric manifold of allowed state transitions. The system is a single, co-evolving entity, a concept with profound implications for fields like reconfigurable computing, artificial life, and the study of any system where the rules of interaction emerge from the state of the system itself.

1.3 Thesis Statement: The Universal Law of Curved Computation

This report develops the formalism for a background-independent theory of computation founded on the following Universal Law:

“The evolution of a computational system follows paths of least informational resistance (geodesics) on a manifold whose curvature is shaped by the local structure of information—its probabilities, causal dependencies, and physical dimensions.”

This principle is captured by two core equations, directly analogous to the field equations and geodesic equation of General Relativity.

The **Computational Field Equation (CFE)** states that the geometry of the computational manifold is determined by its informational content:

Here, \mathcal{G}_{\mu\nu} is the computational Einstein tensor, encoding the manifold's curvature. \mathcal{I}_{\mu\nu} is the information-structure tensor, representing the local density and flux of information. \kappa is a system-dependent constant that couples information to geometry.

The **Computational Geodesic Equation** describes the system's evolution as motion along the "straightest possible paths" within this curved manifold:

Here, x^\alpha(\tau) is the trajectory of the computation through its state space, and \Gamma^\alpha_{\beta\gamma} are the Christoffel symbols that encode the manifold's connection, defining the local rules for "straight" motion.

Together, these equations form a closed, self-consistent system. The information structure determines the geometry, and the geometry determines the evolution of the information structure. The computational "background" is not a fixed stage but is itself a solution to the system's dynamical equations. This establishes a truly background-independent theory of computation, where the laws of evolution are emergent properties of the information itself.

Section 2: The Computational Manifold: The Geometric Space of States

To formalize a geometric theory of computation, one must first define the arena in which computation occurs. This arena is not the physical hardware but the abstract space of all possible states the system can occupy. The Universal Law of Curved Computation identifies this space with a specific mathematical object: a statistical manifold. This choice is not arbitrary; it provides a natural and rigorous way to equip the space of computational states with a geometric structure—including notions of distance, volume, and curvature—derived directly from the principles of information theory and statistics.

2.1 Statistical Manifolds as State Spaces

A computational system, particularly one involving probabilistic elements or uncertainty, can be described at any moment by a probability distribution over its set of possible configurations. A parametric family of such distributions, $p(x|\theta)$, where $\theta = (\theta^1, \dots, \theta^n)$ is a vector of parameters, forms a **statistical manifold**, denoted M. Each point θ on this manifold corresponds to a unique probability distribution, and thus to a unique state of the computational system. A statistical manifold is a smooth, possibly curved space that locally resembles the familiar Euclidean space $\mathbb{R}^n$, allowing the powerful tools of differential geometry to be applied.

Consider a few examples to make this concept concrete:

- **Multinomial Distributions:** A system with m possible discrete outcomes (e.g., the state of a register) is described by a probability vector $\theta = (\theta^1, \dots, \theta^m)$ where $\sum_i \theta^i = 1$ and $\theta^i \ge 0$. This family of distributions forms a statistical manifold of dimension m-1, known as the standard simplex S_{m-1}.
- **Gaussian Distributions:** A system whose state is described by a set of continuous variables with Gaussian noise can be parameterized by its mean μ and covariance Σ. The family of all such Gaussian distributions forms a statistical manifold where each point represents a specific mean and covariance. The state of a learning algorithm, such as the weights of a neural network, can often be modeled as a point on such a manifold.

By identifying the state space of a computation with a statistical manifold, we move from a discrete set of states to a continuous, differentiable space. This allows us to speak of infinitesimal changes in state, trajectories of computation as curves through the manifold, and the local geometry that governs these trajectories.

2.2 The Natural Metric of Information: The Fisher-Rao Metric

A manifold, by itself, is like a rubber sheet; it has a topology but no inherent notion of distance or angle. To define a geometry, one must introduce a **Riemannian metric**, a tensor field g that defines an inner product on the tangent space at each point, allowing for the measurement of

lengths of curves and angles between vectors. For a statistical manifold, there exists a uniquely natural choice for this metric, one that is not imposed externally but arises from the very properties of probability and information: the **Fisher-Rao Information Metric**.
The Fisher-Rao metric, $g_{jk}(\theta)$, can be derived from two equivalent and fundamental perspectives, solidifying its canonical status.

1. **From Statistical Distinguishability:** The distance between two nearby points on the manifold, θ and $\theta + d\theta$, should correspond to how easily one can distinguish the two corresponding probability distributions, $p(x|\theta)$ and $p(x|\theta + d\theta)$, based on an observation x. A natural measure of distinguishability is the variance of the relative change in log-likelihood. This leads directly to the definition of the infinitesimal squared distance $d\ell^2$ as:where the components of the metric tensor are given by the Fisher Information Matrix:This formulation reveals that the geometry of the computational manifold is fundamentally about *distinguishability*. A large distance between two points means the corresponding system states are easily told apart.
2. **From Relative Entropy:** The Kullback-Leibler (KL) divergence, or relative entropy, $D_{KL}(p_1 || p_2)$, is a fundamental measure of the difference between two probability distributions. The Fisher Information Metric can be shown to be the Hessian (matrix of second derivatives) of the KL divergence with respect to the parameters θ. For two infinitesimally close distributions $p(x|\theta)$ and $p(x|\theta+d\theta)$, the KL divergence is: $$D_{KL}(p(x|\theta) || p(x|\theta+d\theta)) \approx \frac{1}{2} \sum_{j,k} g_{jk}(\theta) d\theta^j d\theta^k$$ This directly links the local geometry of the manifold to the foundational concept of information entropy. Curvature, which is derived from second derivatives of the metric, can thus be understood as a measure of the *third-order* change in information content, capturing the non-linear complexity of the state space. A region of high curvature is one where small changes in parameters lead to large, unpredictable changes in the system's behavior, making the relationship between parameters and outcomes highly distorted and complex. Conversely, flat regions correspond to simple, linear, and predictable behavior.

This framework also provides a powerful bridge between discrete and continuous computation. For a discrete probability space, the Fisher metric can be shown to be equivalent to the standard Euclidean metric on the positive orthant of a hypersphere, under the change of variables $u_i = \sqrt{p_i}$. This insight is not merely a mathematical convenience; it demonstrates that the abstract statistical manifold has a concrete geometric embedding, allowing the same fundamental principles to be used in analyzing the geometry of both continuous systems (like neural networks) and discrete systems (like finite automata).

2.3 The Geometric Toolkit: Connection, Transport, and Curvature

With a manifold M and a metric g established, the full machinery of differential geometry becomes available to describe the dynamics of computation.

- **Affine Connection and Christoffel Symbols:** To compare state-change vectors at different points in the manifold, we need a rule for differentiation. An **affine connection** ∇ provides this rule, defining a covariant derivative that describes how a vector field changes along a curve. In a local coordinate system, the connection is fully specified by a set of coefficients $\Gamma^\alpha_{\beta\gamma}$ called the **Christoffel symbols of the second kind**. These symbols are not tensors themselves but encode how the basis vectors change from point to point, defining the "rules of the road" for navigating the manifold.

- **Parallel Transport:** The connection ∇ defines the concept of **parallel transport**: the process of sliding a vector along a curve on the manifold such that its covariant derivative along the curve is zero. In essence, it is the way to move a vector from one point to another without "turning" it, relative to the local geometry. In a computational context, parallel transport describes how a state transition (a tangent vector) is constrained as the system evolves. It defines what it means for a computational process to maintain a "constant direction" in the curved state space.
- **Curvature:** In a flat Euclidean space, parallel transporting a vector around a closed loop returns it to its original orientation. On a curved surface like a sphere, this is not the case. The failure of a vector to return to its original state after being parallel transported around an infinitesimal closed loop is the definition of **curvature**. Curvature is a local, intrinsic property of the manifold that quantifies the degree to which the geometry deviates from being flat. Computationally, it represents the non-commutativity of state transitions. In a flat region, applying update A then update B yields the same result as applying B then A. In a curved region, the order of operations matters profoundly, a hallmark of complex computational systems. The curvature is fully characterized by the Riemann curvature tensor, which is constructed from the Christoffel symbols and their derivatives.

Section 3: The Sources of Curvature: Deconstructing the Information-Structure Tensor ($\mathcal{I}_{\mu\nu}$)

In General Relativity, the Einstein tensor $\mathcal{G}_{\mu\nu}$ represents the geometry of spacetime, while the stress-energy tensor $\mathcal{T}_{\mu\nu}$ represents the matter and energy that act as the source of that geometry. The Computational Field Equation, $\mathcal{G}_{\mu\nu} = \kappa \mathcal{I}_{\mu\nu}$, is built on an identical structure. Having defined the geometric side of the equation, we now turn to its source: the **Information-Structure Tensor, $\mathcal{I}_{\mu\nu}$**. This tensor is the computational analogue of matter. It represents the local density, flux, and structural properties of information that warp the computational manifold, giving rise to curvature. The tensor can be conceptually deconstructed into three fundamental components, corresponding to hierarchical levels of description: the nature of the state itself (probabilistic), the relationships between states (causal), and the rules governing the states (structural).

3.1 Component 1: Probabilistic Divergence (Entropy and Uncertainty)

The first and most fundamental source of curvature is the static distribution of information itself. This component of $\mathcal{I}_{\mu\nu}$ represents the "information mass" or "information density" at a point in the computational manifold. It is related to the concepts of entropy, uncertainty, and the local volume of the state space.
On a Riemannian manifold, the volume element is not uniform but depends on the metric: $dV = \sqrt{\det(g)} \, d^n\theta$. Since the metric g is the Fisher Information Matrix, the local information volume is itself a function of the informational properties of the state. Regions where the parameters are highly sensitive (large Fisher information) have a larger volume element, implying a greater density of distinguishable states. This density of states acts as a source of curvature.

More formally, this component is directly related to the Fisher Information Matrix itself. As established, the Fisher metric g_{jk} is the Hessian of the KL-divergence. The Ricci curvature, a key component of the Einstein tensor $\mathcal{G}_{\mu\nu}$, is constructed from derivatives of the metric. Therefore, the very presence of a non-trivial Fisher metric—that is, a space of probability distributions where parameters have a discernible effect—is a source of curvature. Regions of the manifold corresponding to sharp, low-entropy distributions (high information) or, conversely, regions of high uncertainty where multiple outcomes are nearly equally likely, can both be seen as concentrations of "informational matter" that warp the geometry. This component represents the scalar potential of information, analogous to the energy density (T_{00}) component of the stress-energy tensor.

3.2 Component 2: Causal Asymmetries (Directed Information Flow)

The second component of $\mathcal{I}_{\mu\nu}$ is dynamic and vectorial, representing the flux and directed flow of information within the system. This elevates causality from a mere statistical correlation to a fundamental, physical source of geometry. Just as moving masses generate gravitomagnetic fields in GR, directed causal influences generate a "gravito-causal" curvature in the computational manifold. This component is the direct analogue of the momentum density and stress components of $\mathcal{T}_{\mu\nu}$, and it is responsible for encoding the arrow of computational time into the fabric of the manifold.

The formalism for this component is grounded in the principles of **Information Geometric Causal Inference (IGCI)**. IGCI is based on the postulate of independent mechanisms: for a direct causal relationship $X \to Y$, the distribution of the cause, $P(X)$, and the mechanism transforming the cause into the effect, $P(Y|X)$, are assumed to be independent. This independence is not merely statistical but can be expressed geometrically as an **orthogonality condition in information space**. The violation of this orthogonality in the anti-causal direction ($Y \to X$) creates a fundamental asymmetry. This asymmetry acts as a "stress" in the information manifold, a directed tension that contributes to its curvature. The strength of this causal influence can be quantified precisely by the KL-divergence between the observed joint distribution and a hypothetical distribution where the causal link is severed by intervention.

This concept is further refined by the framework of **Causal Geometry**, which formalizes the relationship between the space of possible interventions on a system and the space of resulting effects. Each space can be endowed with its own metric. An effective and informative causal model is one where the geometry of interventions is well-matched to the geometry of effects. A significant mismatch, or incongruence, between these geometries indicates a strong, non-trivial causal structure that constrains the system's evolution. We propose that this causal incongruence is a primary source term in the Information-Structure Tensor. It represents the directed flow of information, analogous to how momentum density (T^{0i}) in GR represents the flow of mass. This component ensures that the geometry of computation is not static but is shaped by the ongoing, directed processes occurring within the system.

3.3 Component 3: Dimensional and Structural Constraints

The third component of $\mathcal{I}_{\mu\nu}$ accounts for the fixed, axiomatic scaffolding of a computational system. These are the constraints imposed by physical laws, dimensional analysis, data types, or architectural invariants. While the theory is background-independent, any specific instantiation of a computation exists within a context that imposes such constraints. These constraints act as boundaries or submanifolds within the larger, unconstrained state

space, and the geometry of their embedding is a source of curvature.
For instance, a scientific simulation involving physical quantities like mass (kg), length (m), and time (s) is bound by the rules of dimensional homogeneity. The equations governing the system are only valid on a submanifold of the total parameter space where the physical units are consistent. The embedding of this physically-valid submanifold within the larger space of all possible parameter values can induce **extrinsic curvature**, separate from the intrinsic curvature generated by probabilistic and causal factors.
Similarly, in a strongly-typed programming language, the rules of the type system restrict the set of valid operations and state transitions. An integer cannot be treated as a function pointer. This creates impassable "walls" or boundaries in the computational manifold. These boundaries act as potent sources of curvature, making it informationally "costly" or "difficult" to transition between states of incompatible types. These structural constraints are analogous to boundary conditions in physical theories. They do not represent a fixed background in the Newtonian sense, but rather define the arena within which the dynamic, background-independent laws operate for a specific system. They contribute to $\mathcal{I}_{\mu\nu}$ by defining the global topology and boundary structure of the accessible state space.
Together, these three components provide a complete description of the "informational matter" that sources geometric curvature. The Information-Structure Tensor captures not only what a system *is* at a given moment (its probabilistic state), but also what it *does* (its causal dynamics) and what it *cannot do* (its structural constraints).

Section 4: The Laws of Evolution: Field Equation and Geodesic Motion

The Universal Law of Curved Computation is defined by two central equations that govern the interplay between information and geometry. The Computational Field Equation dictates how the structure of information shapes the computational manifold, while the Computational Geodesic Equation describes how a system evolves through that shaped manifold. This section provides a formal analysis of these laws, establishing their mathematical underpinnings and connecting them to concrete, practical principles in machine learning and optimization.

4.1 The Computational Field Equation (CFE): $\mathcal{G}_{\mu\nu} = \kappa \mathcal{I}_{\mu\nu}$

The CFE is the core dynamical law of the theory, establishing the relationship between the geometry of the computational state space and the information contained within it.

- **The Geometric Side ($\mathcal{G}_{\mu\nu}$):** The left side of the equation is the **Computational Einstein Tensor**, $\mathcal{G}_{\mu\nu}$. This tensor is constructed from the metric tensor, $g_{\mu\nu}$, which in this theory is the Fisher-Rao Information Metric. Specifically, $\mathcal{G}_{\mu\nu} = \mathcal{R}_{\mu\nu} - \frac{1}{2}\mathcal{R}g_{\mu\nu}$, where $\mathcal{R}_{\mu\nu}$ is the Ricci curvature tensor and $\mathcal{R}$ is the Ricci scalar. The Ricci tensor is itself a contraction of the full Riemann curvature tensor and represents the change in the volume of a small ball of geodesics, effectively measuring how much the geometry deviates from being Euclidean on average. Thus, $\mathcal{G}_{\mu\nu}$ is a pure encoding of the manifold's geometric properties, derived entirely from the Fisher metric.

- **The Informational Side ($\mathcal{I}_{\mu\nu}$):** The right side of the equation contains the **Information-Structure Tensor**, $\mathcal{I}_{\mu\nu}$, which, as detailed in the previous section, acts as the source of the geometry. It is a comprehensive representation of the system's informational content, including probabilistic divergence, causal flux, and structural constraints.
- **The Coupling Constant (κ):** The constant κ mediates the relationship between information and geometry, analogous to the term $8\pi G/c^4$ in General Relativity. It is a system-dependent constant that can be interpreted as the **computational plasticity** or **learnability** of the system. A system with a large κ is highly "flexible"; even a small amount of informational structure (e.g., a weak causal link) can induce significant curvature, creating a complex and rugged computational landscape. Such systems may be powerful but difficult to optimize. Conversely, a system with a small κ is "rigid" or "stiff"; it requires a massive concentration of information to bend the manifold. These systems tend to remain geometrically simple, making their behavior more predictable and easier to optimize. The value of κ could serve as a fundamental parameter characterizing the intrinsic complexity of a given class of computational problems.

The CFE thus provides a complete statement of the dynamic relationship: the structure and content of information dictate the geometry of the space of possible computations.

4.2 The Principle of Extremal Informational Action: Geodesics as Computational Trajectories

Given a curved manifold defined by the CFE, the second law of the theory specifies how a system evolves within it. The evolution follows a **geodesic**, which is the path of extremal length or "informational action." A geodesic is the generalization of a "straight line" to a curved space. The path $x^\alpha(\tau)$ of a computation is governed by the **Computational Geodesic Equation**:

This equation describes a path of zero acceleration, where the "acceleration" is measured relative to the manifold's intrinsic geometry as encoded by the Christoffel symbols $\Gamma^\alpha_{\beta\gamma}$. The parameter τ represents an abstract "computational proper time," an invariant measure of the computation's progress, which could correspond to the number of execution steps, the amount of entropy produced, or physical clock time, depending on the context.

This abstract physical law has a direct and profound connection to a concrete, state-of-the-art optimization algorithm: **Natural Gradient Descent (NGD)**. This connection bridges the gap between the high-level theory and practical computation.

- **Standard vs. Natural Gradient Descent:** Standard gradient descent algorithms minimize a loss function $L(\theta)$ by iteratively updating parameters in the direction of the negative gradient, $-\nabla L(\theta)$. This direction is the "steepest descent" direction as measured by the standard Euclidean distance in the parameter space. However, this approach is blind to the fact that the parameter space is not uniform; a step of a given Euclidean size in one direction might cause a massive change in the model's output distribution, while a step of the same size in another direction might have a negligible effect.
- **The Geometry of NGD:** Natural Gradient Descent, pioneered by Shun-ichi Amari, corrects this by defining the "steepest descent" direction relative to the intrinsic geometry of the statistical manifold of probability distributions. It takes steps that are of constant

size not in the Euclidean parameter space, but in the space of distributions as measured by the Fisher-Rao metric. The NGD update rule is given by:where $g(\theta_t)^{-1}$ is the inverse of the Fisher Information Matrix. This pre-multiplication by the inverse metric tensor effectively corrects the gradient, accounting for the curvature of the information space and pointing towards the true direction of steepest descent on the manifold.
- **Geodesics as NGD Flow:** The Computational Geodesic Equation is precisely the continuous-time differential equation that describes the trajectory of this Natural Gradient flow. A system evolving according to the Universal Law is, therefore, performing an optimal learning or optimization process. It is continuously following the most efficient path toward a local minimum of informational action. This provides a deep physical justification for the superior performance of NGD in many machine learning tasks. Algorithms like NGD are not merely clever heuristics; they are successful because they respect the intrinsic geometry of the problem space. Standard gradient descent is akin to navigating the curved surface of the Earth with a flat, Euclidean map—it works reasonably well over short distances but fails to capture the global structure. NGD is akin to navigating with a globe, following the great-circle routes that are the true "straight lines" or geodesics of the planet.

This correspondence is summarized in the following table, which details the structural isomorphism between General Relativity and the proposed theory of Curved Computation.

Feature	Physics (General Relativity)	Computation (Curved Computation Theory)
Fundamental Arena	Spacetime Manifold	Statistical (Information) Manifold
Points in Arena	Spacetime Events (x, y, z, t)	Probabilistic States of a System ($p(x
Source of Curvature	Stress-Energy Tensor ($\mathcal{T}_{\mu\nu}$)	Information-Structure Tensor ($\mathcal{I}_{\mu\nu}$)
Content of Source	Energy, Momentum, Pressure, Stress	Probabilistic Divergence, Causal Flux, Structural Constraints
Field Equation	$G_{\mu\nu} = \frac{8\pi G}{c^4} \mathcal{T}_{\mu\nu}$	$\mathcal{G}_{\mu\nu} = \kappa \mathcal{I}_{\mu\nu}$
Law of Motion	Objects follow geodesics (paths of extremal proper time)	Systems evolve along geodesics (paths of extremal informational action)
"Straight Line"	Inertial Motion	Optimal computational path (Natural Gradient flow)
Underlying Principle	Principle of Equivalence / General Covariance	Principle of Computational Equivalence / Background Independence
Dynamical Nature	Mass-energy tells spacetime how to curve; spacetime tells mass-energy how to move.	Information tells the manifold how to curve; the manifold tells the computation how to evolve.

Section 5: The Principle of Background Independence

A central claim of the Universal Law of Curved Computation is that it constitutes a

background-independent theory. This property is not merely an aesthetic preference but represents a radical departure from existing models of computation and carries profound implications for how we understand and engineer information-processing systems. To appreciate its significance, one must first understand what background independence means in its original physical context and then identify the implicit backgrounds that this new theory eliminates.

5.1 Defining Background Independence

In theoretical physics, background independence is a condition requiring that the fundamental equations of a theory do not rely on any pre-supposed, fixed geometric structures. A theory is background-independent if all its geometric entities, such as the metric tensor that defines distances and angles, are dynamical variables that are determined by solving the equations of the theory itself. In such a theory, there is no distinction between the "stage" and the "actors"; the stage is itself an actor, co-evolving with the other elements of the system.

General Relativity is the canonical example of a background-independent theory. The metric of spacetime is not given a priori; it is the solution to the Einstein Field Equations for a given distribution of mass and energy. In stark contrast, Newtonian mechanics and Special Relativity are background-dependent. They are formulated on the fixed, immutable stages of absolute Euclidean space and Minkowski spacetime, respectively. The geometry in these theories is an absolute object, affecting motion but unaffected by it.

5.2 The Backgrounds of Conventional Computation

When viewed through this lens, nearly all conventional models of computation are revealed to be profoundly background-dependent. They are built upon a foundation of fixed, non-dynamical structures that serve as the absolute stage for computational processes.

- **The Turing Machine:** The quintessential model of computation relies on the fixed background of a one-dimensional, infinite tape and a finite-state controller with a fixed transition table. The "geometry" of the computation is static and externally imposed.
- **The Von Neumann Architecture:** Modern computers are physical instantiations of the von Neumann model, which is defined by the background structures of a central processing unit (CPU) with a fixed instruction set and a separate, linearly addressable memory space. The rules of computation are hardwired into the silicon.
- **Programming Languages:** Even at the software level, the syntax and semantics of a programming language form a rigid background. The set of valid operations and data structures is pre-defined, creating a fixed logical space in which algorithms are expressed.

In all these cases, the "arena" of computation is static. The rules of evolution are absolute and do not change in response to the information being processed.

5.3 Emergent Geometry as the Foundation for Independence

The Universal Law of Curved Computation achieves background independence by eliminating these fixed structures. In this theory, there is no pre-ordained computational space or set of rules.

1. **The Manifold is Emergent:** The computational manifold is not a pre-defined data structure. It is the space of all possible probability distributions that can describe the system. Its existence and dimensionality are determined by the system's degrees of

freedom, not by an external specification.

2. **The Geometry is Dynamic:** The metric of the manifold—its entire geometric structure—is not fixed. It is determined dynamically at every moment by the Computational Field Equation, $\mathcal{G}_{\mu\nu} = \kappa \mathcal{I}_{\mu\nu}$. The local and global shape of the computational space is a direct consequence of the system's current informational state.
3. **The Laws of Motion are Emergent:** The "rules" of computation are the available geodesic paths. Since the geometry is dynamic, the geodesics are also dynamic. The set of possible, efficient state transitions is not a fixed instruction set but an emergent property of the manifold's curvature. The theory possesses only dynamical entities; there are no fixed fields or absolute objects.

This framework provides a new language for describing computation in systems where the distinction between program and data is blurred or non-existent. Consider a biological cell: there is no CPU, no instruction set, no separate memory. The "computation" consists of a complex network of chemical reactions. The state of the system (the concentrations of various proteins and metabolites) defines a point on a statistical manifold. The interactions and causal relationships between these molecules (e.g., enzyme kinetics, gene regulation) form the Information-Structure Tensor, $\mathcal{I}_{\mu\nu}$. This tensor, in turn, defines the curvature of the manifold, and the most probable future reaction pathways are the geodesics through this curved chemical space. The Universal Law thus offers a potential framework for a fundamental theory of biological organization and information processing.

This principle also forces a re-evaluation of fundamental concepts like computational "errors" or "bugs." In a traditional, background-dependent system, a bug is a deviation from a fixed, externally-defined specification. It is a failure to follow the correct, pre-ordained path. In a background-independent system, there is no external specification. The system's evolution always follows a geodesic, which is by definition the "natural" and "straightest" path according to its own intrinsic geometry at that moment. An undesirable outcome is not an "error" in this sense, but rather the result of the system following a valid geodesic into a region of the state space that is considered undesirable from an external perspective. This reframes the problem of control and debugging: instead of "fixing the code" (altering the path), one must engage in a form of "informational engineering" or "geometric terraforming." The goal is to modify the information landscape—the sources of curvature in $\mathcal{I}_{\mu\nu}$—to reshape the geodesics themselves, steering them away from undesirable regions and toward desired ones.

Section 6: Recursive Closure: The Co-evolution of Computation and Geometry

The principle of background independence leads to the theory's most profound and complex feature: a recursive, dynamical feedback loop between the computational process and the geometric space in which it unfolds. In General Relativity, mass-energy curves spacetime, and that curvature guides the motion of mass-energy. Similarly, in the Universal Law of Curved Computation, the information structure shapes the geometry of the computational manifold, and that geometry guides the evolution of the information. This creates a closed loop of co-evolution, where the computation and its underlying space continuously and recursively define one another.

6.1 The Dynamical System

The evolution of a system governed by the Universal Law can be formalized as a discrete-time dynamical system. The complete state of the system at any time t is not just its informational content but the pair (θ_t, g_t), where θ_t is the point on the manifold representing the system's probabilistic state, and g_t is the metric tensor defining the geometry of the manifold at that time.
An elementary step in the system's evolution proceeds as follows:

1. **Geodesic Motion:** Given the state (θ_t, g_t), the system evolves for an infinitesimal step of computational time $d\tau$ along the geodesic originating at θ_t. This determines the next informational state, θ_{t+1}.
2. **Update of Information Structure:** The new state θ_{t+1}, along with any changes in causal relationships or external constraints, defines a new Information-Structure Tensor, $\mathcal{I}_{t+1}$.
3. **Solving the Field Equation:** The Computational Field Equation, $\mathcal{G}_{\mu\nu} = \kappa \mathcal{I}_{\mu\nu}$, is conceptually solved for the new geometry. The information structure $\mathcal{I}_{t+1}$ acts as the source term to determine the new metric tensor, g_{t+1}.
4. **New System State:** The system arrives at its new state, (θ_{t+1}, g_{t+1}), and the cycle repeats.

This is a non-linear, coupled dynamical system where the state and the laws governing the evolution of the state are in constant interplay. The computation is not merely exploring a static landscape; it is actively reshaping the landscape with every step it takes.

6.2 Connection to Self-Modifying Systems

This recursive closure provides a deep, geometric foundation for the concept of **self-modifying algorithms and code**. A self-modifying system is one that can alter its own instructions during execution. In the context of our theory, the "instructions" are the geodesics—the paths of least informational resistance. The act of computation—moving along a geodesic—alters the information landscape ($\mathcal{I}_{\mu\nu}$). This change in information structure, via the CFE, alters the manifold's geometry ($g_{\mu\nu}$). A change in geometry, in turn, alters the set of available geodesics for the next computational step.
Therefore, the system is continuously "rewriting" its own instruction set. This is not a superficial modification, like changing a value in memory, but a fundamental reconfiguration of the possible paths of evolution. The theory describes a form of intrinsic, geometric self-modification where the distinction between execution and compilation, or between data and program, dissolves completely. Every computational step is simultaneously an act of execution and an act of modifying the laws that govern future executions.

6.3 Stability and Emergence

The long-term behavior of such a recursively closed system raises fundamental questions about stability, complexity, and emergence. Does the co-evolution of state and geometry lead to chaotic, unpredictable behavior, or can it converge to stable structures? This framework provides a mechanism for understanding how complex, stable computational ecosystems might emerge from simple initial conditions. A system might initially exist in a simple, nearly flat geometry. As it computes, it might generate information structures that create localized pockets of curvature. These curved regions could act as "attractors," channeling future computational trajectories and reinforcing their own structure. Over time, this feedback loop could lead to the emergence of highly complex, stable geometric features that correspond to sophisticated computational functions.

This process offers a physical mechanism for open-ended evolution and the creation of genuine novelty. In a typical evolutionary algorithm, agents explore a fixed fitness landscape. In this theory, the agent (the computational state) and the landscape (the information manifold) are coupled. The agent's movement deforms the landscape, which in turn changes the "fitness gradients" that guide future movement. This allows the system to create new computational possibilities—new valleys, hills, and channels in the manifold—that did not exist in the initial geometry. It is a model for niche construction, where an organism actively shapes its environment, thereby altering the very selection pressures to which it is subject.

Furthermore, this recursive structure suggests a novel model for memory. In a conventional system, memory is stored data—a specific point \theta on the manifold. In this theory, memory can also be encoded in the *geometry* of the manifold itself. A significant computational event can leave a persistent "dent" or "scar" in the manifold's curvature. This geometric alteration will influence all future computational trajectories that pass through that region, even if the explicit data representing the event (\theta) is long gone. Memory becomes a physical trace left in the fabric of the computational space, analogous to how a river carves a canyon into a landscape, making it ever more probable that future flows of water will follow the same path. This geometric memory is non-local and persistent, providing a physical basis for how past experiences can shape future processing in a fundamental way.

Section 7: Implications, Applications, and Future Horizons

The Universal Law of Curved Computation, by unifying principles from physics, computer science, and information theory, opens up new avenues of research and offers a novel lens through which to view some of the most profound challenges in science. Its implications extend from the theoretical foundations of computational complexity to the practical design of next-generation algorithms and adaptive systems. This final section explores these broader consequences, outlining a path from the abstract continuous theory to concrete applications and sketching the future horizons of a true physics of information.

7.1 A Geometric Re-framing of Computational Complexity

The theory of computational complexity, which seeks to classify problems based on the resources required to solve them, has long sought a deeper, more physical foundation. The Universal Law provides a natural geometric language for this endeavor, connecting directly with existing research programs like **Geometric Complexity Theory (GCT)**. GCT aims to resolve major open questions, such as the P vs. NP problem, by using advanced tools from algebraic geometry and representation theory to study the symmetries of computational problems. Our framework provides a physical interpretation for these geometric structures.

We hypothesize that computational complexity classes correspond to distinct geometric and topological properties of the information manifolds generated by problems within those classes.

- **P Problems:** Problems solvable in polynomial time might generate information manifolds that are geometrically "simple." These manifolds may possess low average curvature, simple topology (e.g., being contractible to a point), or other properties that allow for the efficient computation of geodesics between any two points. The path to a solution is relatively straight and unobstructed.

- **NP-hard Problems:** Problems for which solutions are difficult to find but easy to verify might generate manifolds with highly complex geometries. These spaces could be characterized by high or rapidly fluctuating curvature, creating a rugged landscape where finding the shortest geodesic (the optimal solution) is an exponentially difficult search problem. The computational complexity of an algorithm, often measured in terms of time steps like O(n \log n) versus O(n^2), could be directly related to the integrated length of the geodesic it traverses on the corresponding manifold.

This perspective may offer a new angle on the P vs. NP problem itself, reframing it as a question of geometry versus topology. The difficulty of finding a solution to an NP problem could be a *geometric* challenge: navigating a complex manifold to find a specific geodesic path. The ease of verifying a solution, on the other hand, could be a *topological* property. Verifying a proposed solution might be equivalent to checking a simple topological invariant, such as whether the start and end points of the proposed path lie within the same connected component of a solution submanifold, a question that could be answered far more easily than the geometric search. This suggests a concrete research program: can the P vs. NP question be formally rephrased as, "For the class of manifolds generated by NP problems, is the geometric problem of finding a specific geodesic computationally harder than the topological problem of determining if a path exists at all?"

7.2 From the Continuum to the Discrete: A Path to Application

The theory presented in this report is formulated in the language of smooth, continuous differential geometry. However, practical computation, as performed by digital computers, is inherently discrete. To bridge this gap and translate the Universal Law into practical algorithms, the essential toolkit is **Discrete Differential Geometry (DDG)**.

DDG is a vibrant field that aims not merely to approximate smooth geometry but to build a consistent discrete analogue of the entire theory. Instead of smooth surfaces, it deals with polygonal meshes and simplicial complexes. The core philosophy of DDG emphasizes a "mimetic" approach, where discrete definitions are carefully constructed to exactly preserve the fundamental structural properties and invariants (like total curvature or conservation laws) of the corresponding smooth theory, regardless of the coarseness of the discretization.

Using the methods of DDG, one can define discrete versions of the Fisher metric, the connection, curvature, and geodesics on the discrete state spaces of digital computations. This provides a principled pathway for designing novel "geometric algorithms" that directly implement the dynamics of the Universal Law. Such algorithms would be inherently adaptive, navigating the discrete problem space by respecting its underlying information geometry.

This approach also suggests a new paradigm for algorithm design, which might be termed **"geometric programming."** In conventional programming, one designs an explicit sequence of operations—a specific path. In geometric programming, one would instead focus on designing the Information-Structure Tensor, \mathcal{I}_{\mu\nu}. The programmer's task would be to specify the desired information landscape: defining the sources of informational "mass," establishing the causal relationships that create "stress," and imposing the constraints that form the "boundaries." The optimal algorithm—the geodesic path—would then emerge automatically as a solution to the laws of motion within that user-defined geometry. This represents a shift from a procedural to a declarative, physics-based paradigm, analogous to an engineer designing a complex gravitational lens to steer light beams along desired paths rather than attempting to program the trajectory of each individual photon.

7.3 Conclusion: Towards a Physics of Information

The Universal Law of Curved Computation offers a synthesis of ideas from General Relativity, information theory, and computer science. It proposes a fundamental shift in perspective: to view computation not as the abstract manipulation of symbols, but as a physical process governed by universal geometric laws. In this framework, the space of computation is a dynamic entity, shaped by the information it contains. The evolution of a system is an optimal, inertial path through this curved space, a principle that finds its concrete expression in algorithms like Natural Gradient Descent. The theory is inherently background-independent and recursively closed, providing a natural language for describing the co-evolutionary dynamics of complex adaptive systems, from biological cells to artificial intelligence.

By grounding computational complexity in the geometry of information manifolds and providing a bridge to practical application via discrete differential geometry, this framework lays the groundwork for a new science of information—one that seeks to uncover the physical laws governing the behavior of intelligent, evolving, and complex systems. It moves us closer to a unified understanding of the universe, not just as a collection of matter and energy, but as a vast, self-organizing computational process, whose very fabric is shaped by the flow and structure of information itself.

Works cited

1. en.wikipedia.org, https://en.wikipedia.org/wiki/Discrete_differential_geometry#:~:text=Discrete%20differential%20geometry%20is%20the,geometry%20processing%20and%20topological%20combinatorics. 2. Background independence - Wikipedia, https://en.wikipedia.org/wiki/Background_independence 3. James Read, Background Independence in Classical and Quantum Gravity | BJPS Review of Books - British Society for the Philosophy of Science, https://www.thebsps.org/reviewofbooks/de-haro-on-read/ 4. General relativity - Wikipedia, https://en.wikipedia.org/wiki/General_relativity 5. A tale of analogies... - arXiv, https://arxiv.org/pdf/2304.02167 6. Reformulation of general relativity brings it closer to Newtonian physics, https://physicsworld.com/a/reformulation-of-general-relativity-brings-it-closer-to-newtonian-physics/ 7. The Basics of Information Geometry - Free, http://djafari.free.fr/MaxEnt2014/papers/Tutorial2_paper.pdf 8. Information geometry in optimization, machine learning and statistical inference, https://bsi-ni.brain.riken.jp/database/file/303/308.pdf 9. Applications of Information Geometry to Machine Learning - Jason d'Eon, https://www.jasondeon.com/files/masters_project.pdf 10. What is the Fisher-Rao distance? - Kisung You, https://www.kisungyou.com/Blog/blog_001_FisherRao.html 11. Fisher information metric - Wikipedia, https://en.wikipedia.org/wiki/Fisher_information_metric 12. Fisher information - Wikipedia, https://en.wikipedia.org/wiki/Fisher_information 13. An Elementary Introduction to Information Geometry - MDPI, https://www.mdpi.com/1099-4300/22/10/1100 14. An Elementary Introduction to Information Geometry - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC7650632/ 15. arxiv.org,

https://arxiv.org/abs/1402.2499#:~:text=Information%20Geometric%20Causal%20Inference%20(IGCI,of%20orthogonality%20in%20information%20space. 16. (PDF) Justifying Information-Geometric Causal Inference, https://www.researchgate.net/publication/260147512_Justifying_Information-Geometric_Causal_Inference 17. Information-geometric approach to inferring causal directions | Empirical Inference, https://is.mpg.de/ei/publications/janzingmzlzdss2012 18. Quantifying causal influences - arXiv, https://arxiv.org/pdf/1203.6502 19. (PDF) Quantifying causal influences - ResearchGate, https://www.researchgate.net/publication/221966206_Quantifying_causal_influences 20. Causal Geometry - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC7824647/ 21. Causal Geometry - DSpace@MIT, https://dspace.mit.edu/handle/1721.1/131313 22. [2010.09390] Causal Geometry - arXiv, https://arxiv.org/abs/2010.09390 23. Natural gradients - Andy Jones, https://andrewcharlesjones.github.io/journal/natural-gradients.html 24. It's Only Natural: An Excessively Deep Dive Into Natural Gradient Optimization - Medium, https://medium.com/data-science/its-only-natural-an-excessively-deep-dive-into-natural-gradient-optimization-75d464b89dbb 25. Full article: Natural Gradient Variational Bayes Without Fisher Matrix Analytic Calculation and Its Inversion - Taylor & Francis Online, https://www.tandfonline.com/doi/full/10.1080/01621459.2024.2392904 26. Fisher Information and Natural Gradient Learning of Random Deep Networks - arXiv, https://arxiv.org/abs/1808.07172 27. Why Natural Gradient?, http://www.yaroslavvb.com/papers/amari-why.pdf 28. A note on the natural gradient and its connections with the Riemannian gradient, the mirror descent, and the ordinary gradient - Frank Nielsen, https://franknielsen.github.io/blog/NaturalGradientConnections/NaturalGradientConnections.pdf 29. Exact natural gradient in deep linear networks and its application to the nonlinear case, http://papers.neurips.cc/paper/7834-exact-natural-gradient-in-deep-linear-networks-and-its-application-to-the-nonlinear-case.pdf 30. Part IV: Natural Gradient Descent and its Extension—Riemannian Gradient Descent - Wu Lin, https://yorkerlin.github.io/posts/2021/11/Geomopt04/ 31. natural gradient in wasserstein statistical manifold, https://people.math.sc.edu/wuchen/papers/ChenLi.pdf 32. Natural Gradient Flow in the Mixture Geometry of a Discrete Exponential Family - MDPI, https://www.mdpi.com/1099-4300/17/6/4215 33. Self-Modifying Algorithms Overview - Emergent Mind, https://www.emergentmind.com/topics/self-modifying-algorithms 34. Self-modifying code - Wikipedia, https://en.wikipedia.org/wiki/Self-modifying_code 35. What is Self-modifying Code? The Evolution of Code Obfuscation - ReasonLabs Cyberpedia, https://cyberpedia.reasonlabs.com/EN/self-modifying%20code.html 36. Self-modifying code - Semantic Scholar, https://www.semanticscholar.org/topic/Self-modifying-code/461812 37. Geometric complexity theory - Wikipedia, https://en.wikipedia.org/wiki/Geometric_complexity_theory 38. Geometry and Complexity Theory - Cambridge University Press & Assessment, https://www.cambridge.org/core/books/geometry-and-complexity-theory/15E3ABA3FF14E1054574663F60250D80 39. Introduction to geometric complexity theory - DCS - Department of Computer Science, https://www.dcs.warwick.ac.uk/~u2270030/teaching_sb/summer17/introtogct/gct.pdf 40. Computational Complexity and Algebraic Geometry, https://www.bimsa.cn/research_detail/ComComandAlgGeo.html 41. [2011.07601] Geometry of quantum complexity - arXiv, https://arxiv.org/abs/2011.07601 42. On complexity and information geometry - Rising Entropy, https://risingentropy.com/on-complexity-and-information-geometry/

43. Information Geometry, Complexity Measures and Data Analysis - MDPI, https://www.mdpi.com/1099-4300/24/12/1797 44. Information Geometry on Complexity and Stochastic Interaction - MDPI, https://www.mdpi.com/1099-4300/17/4/2432 45. Computational geometry - Wikipedia, https://en.wikipedia.org/wiki/Computational_geometry 46. Discrete differential geometry - Wikipedia, https://en.wikipedia.org/wiki/Discrete_differential_geometry 47. A Glimpse into Discrete Differential Geometry, https://www.ams.org/notices/201710/rnoti-p1153.pdf