Karp Algebraic Reduction Manifold Architecture (KARMA): A Geometric Framework for NP-Complete Problem Equivalence

Basker Palaniswamy *

Posted Date: 11 March 2026

doi: 10.20944/preprints202603.0871.v1

Keywords: NP-completeness; Karp's 21 problems; geometric complexity theory; manifold theory; equivalence classes; polynomial-time reductions; single-reduction propagation

 doi:10.20944/preprints202603.0871.v1

Article

Karp Algebraic Reduction Manifold Architecture (KARMA): A Geometric Framework for NP-Complete Problem Equivalence

Basker Palaniswamy

Insight Research Ireland Centre for Data Analytics, University College Cork (UCC), Cork City, Ireland, European Union; basker170889@gmail.com; basker170889@zohomail.eu

Abstract

In 1972, computer scientist Richard Karp made a remarkable discovery: twenty-one very different problems—from routing networks and planning schedules to packing items efficiently—are all equally difficult in a deep mathematical sense. These problems are now called *NP-complete*, and for more than fifty years researchers have shown their connection by carefully transforming one problem into another step by step. While this approach proves that the problems are related, it often hides the bigger picture of why they share the same level of difficulty. This paper proposes a new way of understanding these problems through geometry. We introduce the **Karp Algebraic Reduction Manifold Architecture (KARMA)**, a framework that places all 21 problems inside a single mathematical "landscape." In this landscape, each problem describes a different region of the same terrain of computational difficulty, and moving from one problem to another becomes like traveling smoothly across this terrain. The framework naturally groups the problems into three families—graph-theoretic, set-theoretic, and number-theoretic problems. In this geometric interpretation, distances represent how difficult it is to transform one problem into another, while the curvature of the landscape reflects their inherent computational hardness. By revealing this hidden geometric structure, the KARMA framework provides a new perspective on computational complexity. Instead of studying hard problems individually, researchers can explore the entire landscape of computational difficulty at once, potentially inspiring new algorithms, better hardness predictions, and intelligent systems that can automatically reason about problem transformations.

Keywords: NP-completeness; Karp's 21 problems; geometric complexity theory; manifold theory; equivalence classes; polynomial-time reductions; single-reduction propagation

---

1. Introduction

This section introduces the central question of the paper: if all NP-complete problems are computationally equivalent, is there a single geometric object—a "shape"—that captures this equivalence? We motivate the KARMA framework by identifying limitations of the traditional reduction-based view and explaining why a manifold (geometric surface) is the right mathematical structure to unify Karp's 21 problems.

The theory of NP-completeness, formally established by Cook [1] and extensively developed by Karp [2], represents one of the most profound discoveries in theoretical computer science. Karp's identification of 21 diverse problems—spanning graph theory, set theory, number theory, and combinatorial optimization—as NP-complete demonstrated the remarkable structural unity underlying computational hardness. However, despite nearly five decades of research, a fundamental question remains unresolved: *What is the natural geometric structure that unifies these problems as members of a true equivalence class?*

 doi:10.20944/preprints202603.0871.v1

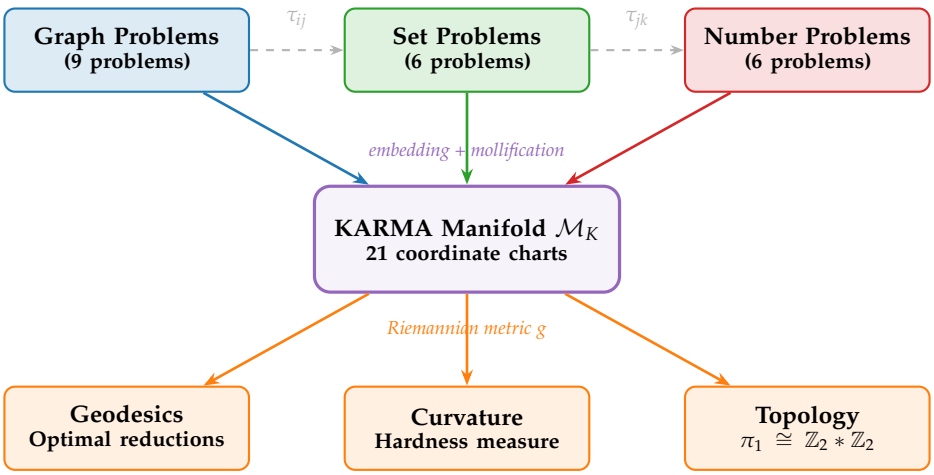

Figure 1. High-level architecture of the KARMA framework. Karp's 21 problems, partitioned into three categories, are embedded as coordinate charts on a unified differentiable manifold. The Riemannian metric yields geodesics (optimal reduction paths), curvature (hardness measure), and topological invariants.

1.1. The Incompleteness of Traditional Reductions

We explain why the classical approach to NP-completeness—proving that one problem can be translated into another in polynomial time—is powerful but has blind spots. It proves equivalence but does not reveal the "shape" of that equivalence.

The standard methodology for establishing NP-completeness relies on polynomial-time many-one reductions (Karp reductions) [3]. To prove a problem Π is NP-complete, one demonstrates:

(i) $\Pi \in$ NP (polynomial-time verification), and
(ii) $\Pi' \leq_p \Pi$ for some known NP-complete problem Π'.

While this approach is computationally elegant, it exhibits several theoretical limitations.

Asymmetry. Proving $\Pi_i \leq_p \Pi_j$ does not immediately yield a natural inverse transformation $\Pi_j \leq_p \Pi_i$, despite their equivalence under NP-completeness.

Non-transitivity in practice. Although reductions compose mathematically, the concatenation of reductions $\Pi_i \to \Pi_j \to \Pi_k$ often obscures the direct structural relationship between Π_i and Π_k.

Absence of canonical form. Unlike algebraic structures with normal forms (e.g., Jordan canonical form for matrices), NP-complete problems lack a distinguished representative to which all others naturally reduce.

Category fragmentation. Karp's 21 problems naturally partition into graph-theoretic (CLIQUE, VERTEX COVER, HAMILTONIAN CYCLE), set-theoretic (SET PACKING, EXACT COVER, SET PARTITION), and number-theoretic (KNAPSACK, PARTITION) categories [4]. Traditional reductions treat these as independent, without recognizing higher-order categorical structure.

1.2. Motivation: From Reductions to Manifolds

We explain the key conceptual leap of this paper: since polynomial-time reducibility is an equivalence relation (every NP-complete problem can be converted to every other), the set of all NP-complete problems forms an equivalence class. In mathematics, equivalence classes naturally carry geometric structure—they can be viewed as surfaces or "manifolds." This is the seed of the KARMA idea.

The central insight motivating this work is that *polynomial-time reducibility defines an equivalence relation, and equivalence classes admit natural geometric interpretations as manifolds*. If we view each NP-complete problem as defining a computational space with inherent dimensionality determined by its input encoding, then the collection of all NP-complete problems should constitute coordinate charts on a unified manifold structure.

 doi:10.20944/preprints202603.0871.v1

This perspective draws inspiration from Geometric Complexity Theory (GCT) [5], category theory applied to computational semantics [6], the differential topology of smooth manifolds [7], and information geometry [8].

1.3. The KARMA Framework

We formally define the central mathematical object of this paper: the KARMA manifold. This is a geometric surface where each of Karp's 21 problems corresponds to a "coordinate chart" (a local description), and the known polynomial-time reductions between problems become smooth transitions between these charts. The manifold is also equipped with a distance measure that encodes how computationally expensive it is to transform one problem into another.

We propose the **Karp Algebraic Reduction Manifold Architecture (KARMA)**, whose central object is defined as follows.

Definition 1 (KARMA Manifold). *Let $\mathcal{K} = \{\Pi_1, \Pi_2, \ldots, \Pi_{21}\}$ denote Karp's 21 NP-complete problems. The KARMA manifold $\mathcal{M}_K$ is a differentiable manifold of dimension d equipped with:*

(a) An atlas $\mathcal{A} = \{(U_i, \phi_i) : i = 1, \ldots, 21\}$ where each chart (U_i, ϕ_i) corresponds to problem Π_i.
(b) Transition functions $\tau_{ij} : \phi_i(U_i \cap U_j) \to \phi_j(U_i \cap U_j)$ representing natural transformations between problems.
(c) A Riemannian metric g encoding computational complexity as geometric distance.
(d) Three distinguished submanifolds $\mathcal{M}_G$, $\mathcal{M}_S$, $\mathcal{M}_N$ corresponding to graph-theoretic, set-theoretic, and number-theoretic problem classes.

Think of each NP-complete problem as a different "coordinate system" for describing the same underlying computational landscape, much as latitude–longitude and UTM are different coordinate systems for describing the same surface of the Earth. The KARMA manifold is that underlying landscape, and the chart maps ϕ_i are the coordinate systems.

1.4. Contributions and Organization

This paper makes the following contributions:

C1. **Foundational framework**: We establish KARMA as a rigorous mathematical structure, proving it satisfies the manifold axioms (Section 3).
C2. **Categorical decomposition**: We prove the three-way split is geometrically natural (Section 4).
C3. **Transformation functions**: We derive explicit smooth transition functions with Jacobian analysis (Section 5).
C4. **Metric structure**: We construct a Riemannian metric with curvature analysis and geodesic characterization (Section 6).
C5. **Computational framework**: We develop practical tools and applications (Section 7).
C6. **Implications**: We discuss consequences for complexity theory, including P vs NP (Section 9).
C7. **Single-reduction propagation**: We prove that a reduction to any one NP-complete problem propagates through the manifold to yield reductions to all 21, formalizing the "reduce once, reach all" principle (Section 8).

2. Preliminaries

Before constructing the KARMA manifold, we need three ingredients: (1) a clear list of Karp's 21 problems, (2) the basic vocabulary of manifold theory from differential geometry, and (3) a precise definition of polynomial-time reductions. This section provides all three, with plain-language explanations alongside the formal definitions. Readers already familiar with complexity theory and differential geometry may skim this section; readers new to either topic will find all the necessary background here.

 doi:10.20944/preprints202603.0871.v1

2.1. Karp's 21 NP-Complete Problems

We list all 21 problems from Karp's seminal 1972 paper, grouped into three natural families. Graph-theoretic problems ask questions about networks (e.g., "Does this network contain a tightly connected group of k nodes?"). Set-theoretic problems ask about collections of objects (e.g., "Can we pick non-overlapping groups that cover everything?"). Number-theoretic problems ask about sums and numerical constraints (e.g., "Can we split these numbers into two groups with the same total?"). This three-way grouping will later correspond to three regions of the KARMA manifold.

We recall Karp's original 21 problems [2], organized by category.

Graph-Theoretic (9 problems): SAT, CLIQUE, VERTEX COVER, HAMILTONIAN CYCLE, HAMILTONIAN PATH, CHROMATIC NUMBER, EXACT COVER, 3-SAT, DIRECTED HAMILTONIAN CYCLE.

Set-Theoretic (6 problems): SET PACKING, FEEDBACK ARC SET, FEEDBACK VERTEX SET, HITTING SET, SET SPLITTING, STEINER TREE.

Number-Theoretic (6 problems): 0-1 INTEGER PROGRAMMING, KNAPSACK, PARTITION, MAX CUT, JOB SEQUENCING, BIN PACKING.

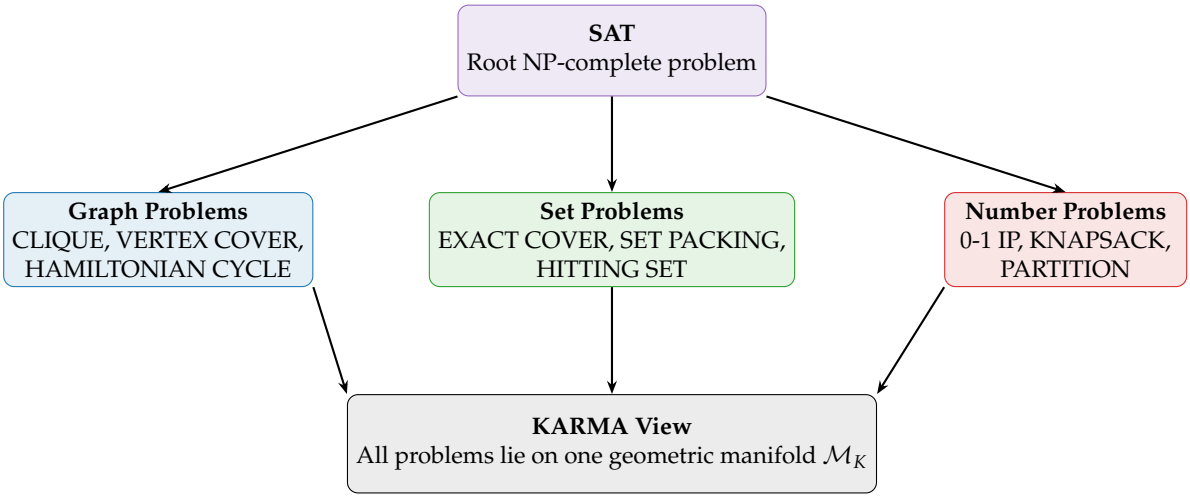

Figure 2. Simplified structure of Karp's NP-complete problems. SAT acts as the common source. Problems naturally group into graph, set, and number categories. In the KARMA framework, these categories correspond to regions of a unified geometric manifold $\mathcal{M}_K$.

2.2. Manifold Theory Essentials

We introduce the mathematical language of manifolds. A manifold is a space that may have a complicated global shape (like the surface of the Earth) but locally looks like ordinary flat space (like a city map). The key objects are *charts* (local maps), *transition functions* (instructions for converting between overlapping maps), and *metrics* (rulers for measuring distance). These concepts will be applied to NP-complete problems in Section 3.

We recall standard definitions from differential geometry [7,9].

Definition 2 (Smooth Manifold). *A smooth manifold of dimension d is a second-countable Hausdorff topological space M equipped with a maximal smooth atlas $\{(U_\alpha, \phi_\alpha)\}$, where each $\phi_\alpha : U_\alpha \to \mathbb{R}^d$ is a homeomorphism, and all transition functions $\tau_{\alpha\beta} = \phi_\beta \circ \phi_\alpha^{-1}$ are C^∞ diffeomorphisms.*

A smooth manifold is a space that can be described by overlapping "maps" (charts), and wherever two maps overlap, the conversion between them is perfectly smooth—no jumps, no corners. The requirement that the space is Hausdorff and second-countable ensures it is well-behaved (distinct points can be separated, and the topology is not "too large" to work with). In our setting, each NP-complete problem will provide one such map.

 doi:10.20944/preprints202603.0871.v1

Definition 3 (Riemannian Manifold). *A Riemannian manifold (M, g) is a smooth manifold equipped with a smoothly varying positive-definite inner product $g_p : T_pM \times T_pM \to \mathbb{R}$ on each tangent space.*

A manifold is a space that "looks like" ordinary Euclidean space when you zoom in. The charts are the "zoomed-in views," and the transition functions translate between overlapping views. A Riemannian metric adds the ability to measure distances and angles.

2.3. Polynomial-Time Reductions: Formal Framework

We define precisely what it means to "convert" one computational problem into another in polynomial time, and prove two key properties: (1) reductions compose (if A converts to B and B converts to C, then A converts to C), and (2) among NP-complete problems, every pair is inter-convertible. These two facts together establish that NP-complete problems form an equivalence class—the algebraic prerequisite for building a manifold.

Definition 4 (Karp Reduction). *A Karp reduction from Π_i to Π_j is a polynomial-time computable function $f_{ij} : \Sigma^* \to \Sigma^*$ such that $x \in \Pi_i \iff f_{ij}(x) \in \Pi_j$.*

A Karp reduction is a translator: given any instance of problem Π_i, it efficiently (in polynomial time) produces an instance of problem Π_j that has a solution if and only if the original did. The translator never needs to solve either problem—it only reformulates the question.

Lemma 1 (Transitivity of Karp Reductions). *If $\Pi_i \leq_p \Pi_j$ via f_{ij} and $\Pi_j \leq_p \Pi_k$ via f_{jk}, then $\Pi_i \leq_p \Pi_k$ via $f_{ik} = f_{jk} \circ f_{ij}$.*

Reductions chain together. If you can translate language A into language B, and language B into language C, then you can translate A into C by applying both translations in sequence. The total translation is still efficient (polynomial time), because the composition of two polynomials is still a polynomial.

Proof. The composition runs in time $O(n^{ab})$ (polynomial), and $x \in \Pi_i \iff f_{ij}(x) \in \Pi_j \iff f_{jk}(f_{ij}(x)) \in \Pi_k$. □

The proof has two parts: (i) the composed function $f_{jk} \circ f_{ij}$ runs in polynomial time because composing polynomials gives a polynomial, and (ii) correctness follows from chaining the two "if and only if" guarantees. This seemingly simple result is the algebraic engine behind the manifold's cocycle condition (Theorem 3).

Lemma 2 (NP-Complete Equivalence). *For any two NP-complete problems Π_i, Π_j, both $\Pi_i \leq_p \Pi_j$ and $\Pi_j \leq_p \Pi_i$ hold.*

Any NP-complete problem can be efficiently translated into any other NP-complete problem, *and back*. This is the defining property of NP-completeness: these problems are all "equally hard" in the sense that solving any one of them efficiently would let you solve all of them efficiently.

Proof. Both are in NP and both are NP-hard, so each reduces to the other [3]. □

The proof is short because it follows directly from the definition of NP-completeness. Being NP-complete means (a) the problem is in NP (solutions can be verified quickly) and (b) it is NP-hard (every NP problem reduces to it). Since both Π_i and Π_j satisfy (a) and (b), each reduces to the other.

Lemmas 1–2 establish that polynomial-time reducibility among NP-complete problems is reflexive, symmetric, and transitive—an equivalence relation. This is the algebraic foundation upon which we build the geometric manifold.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 March 2026 doi:10.20944/preprints202603.0871.v1

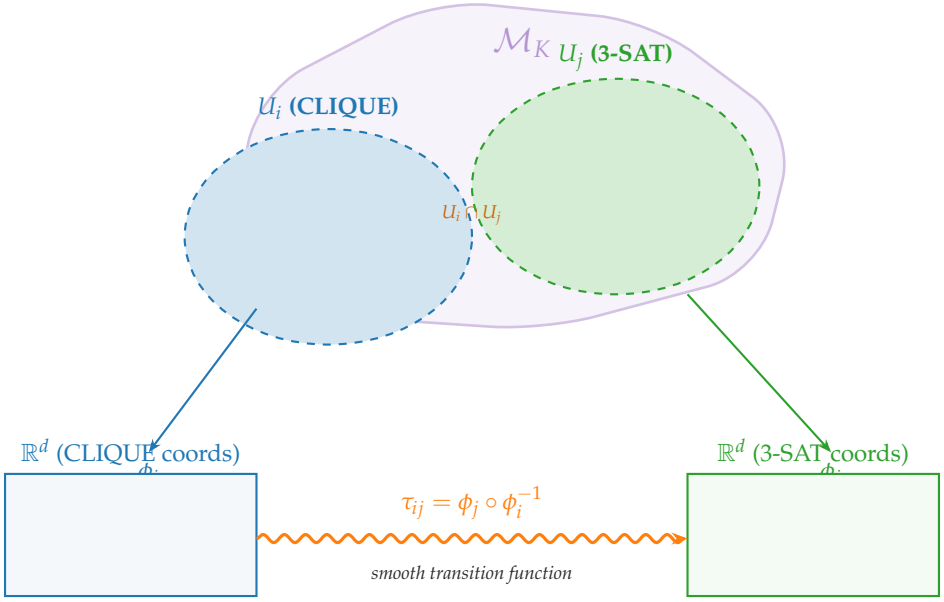

Figure 3. Coordinate charts on the KARMA manifold. Each NP-complete problem defines a chart (U_i, ϕ_i) mapping a neighbourhood of $\mathcal{M}_K$ into $\mathbb{R}^d$. In the overlap $U_i \cap U_j$, the transition function τ_{ij} is constructed from the polynomial-time reduction, smoothed via mollification.

3. Construction of the KARMA Manifold

This is the technical heart of the paper. We show how to build the KARMA manifold step by step: (1) encode each problem instance as a point in Euclidean space, (2) normalize all encodings to live in the same ambient space, (3) define coordinate charts, (4) construct smooth transition functions between charts using mollification, and (5) verify the manifold axioms. By the end of this section, we will have a rigorously defined smooth manifold $\mathcal{M}_K$ with 21 coordinate charts.

3.1. Encoding Problems as Euclidean Structures

The first step in building a geometric object from computational problems is to represent each problem instance as a point in ordinary Euclidean space $\mathbb{R}^d$. We show that every NP-complete problem instance—whether it is a graph, a collection of sets, or a list of numbers—can be uniquely encoded as a vector of real numbers. Different problems may require different numbers of coordinates (a graph on n vertices needs n^2 entries for its adjacency matrix, while a knapsack with n items needs only $2n + 3$ numbers), but each encoding is faithful: no information is lost.

Definition 5 (Problem Embedding)**.** *For problem Π_i with characteristic parameters $(n_1, \ldots, n_m)$, the embedding map is $\iota_i : \text{Instance}(\Pi_i) \to \mathbb{R}^{d_i}$, where d_i is the ambient dimension.*

For each of the 21 problems, we define a function ι_i that takes a problem instance (e.g., a graph together with a number k) and outputs a list of numbers (a vector in $\mathbb{R}^{d_i}$). The length d_i of this list depends on the problem type and the instance size. This is analogous to how a photograph can be represented as a list of pixel values—the image itself is a complex object, but its digital encoding is just a long list of numbers.

Example 1 (Representative Embeddings)**.**
CLIQUE: $\iota_{\text{CLQ}} : (G, k) \mapsto (\text{vec}(A_G), k, n) \in \mathbb{R}^{n^2+2}$.
3-SAT: $\iota_{\text{3SAT}} : \phi \mapsto (\mathbf{c}_1, \ldots, \mathbf{c}_m, n, m) \in \mathbb{R}^{3m+2}$.
KNAPSACK: $\iota_{\text{KS}} : (w, v, W, V) \mapsto (w_1, \ldots, w_n, v_1, \ldots, v_n, W, V, n) \in \mathbb{R}^{2n+3}$.

Lemma 3 (Injectivity of Embeddings)**.** *Each embedding ι_i is injective.*

 doi:10.20944/preprints202603.0871.v1

No two different problem instances are mapped to the same point. In other words, the encoding is lossless—you can always reconstruct the original problem from its vector representation. This is essential: if two different instances mapped to the same point, the geometric structure would confuse them, and the manifold would lose information.

Proof. The embedding encodes all information needed to reconstruct the instance. For CLIQUE, the full adjacency matrix plus k and n uniquely determines (G, k). The argument is analogous for all problems in $\mathcal{K}$. □

The proof observes that each embedding includes every piece of data that defines the instance: for CLIQUE, this means the entire adjacency matrix (which completely describes the graph) plus the parameters k and n. Since all defining data is present, distinct instances produce distinct vectors.

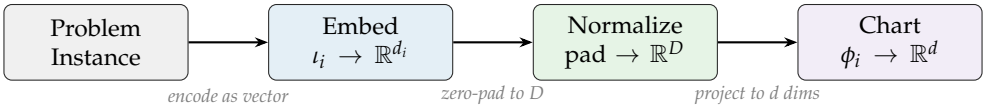

Figure 4. Embedding pipeline. Each problem instance is encoded as a vector, zero-padded to a common ambient dimension D, then projected to the intrinsic manifold dimension d.

3.2. Normalization

Different problems have different embedding dimensions (CLIQUE needs $n^2 + 2$ numbers, KNAPSACK needs $2n + 3$). To place them all in the same ambient space, we "zero-pad" shorter vectors—like adding trailing zeros to a zip code so that all codes have the same length. This normalization does not distort distances; it is a mathematically perfect embedding.

Definition 6 (Normalized Embedding). *For each Π_i with dimension d_i, the normalized embedding into $D = \max_i d_i$ is $\hat{\iota}_i(x) = (\iota_i(x), 0, \ldots, 0) \in \mathbb{R}^D$.*

We pick the largest embedding dimension D among all 21 problems, and then extend every shorter vector to length D by appending zeros. This places all problem instances into a common ambient space $\mathbb{R}^D$ without changing any distances.

Lemma 4 (Isometric Embedding). *The padding map is an isometric embedding of $(\mathbb{R}^{d_i}, \|\cdot\|_2)$ into $(\mathbb{R}^D, \|\cdot\|_2)$.*

Adding zeros to a vector does not change the distance between any pair of vectors. If two CLIQUE instances were distance 5 apart in $\mathbb{R}^{n^2+2}$, they remain distance 5 apart in the larger space $\mathbb{R}^D$. This guarantees that the normalization step introduces no geometric distortion.

Proof. $\|\hat{\iota}_i(x) - \hat{\iota}_i(y)\|_2 = \sqrt{\|\iota_i(x) - \iota_i(y)\|_2^2 + 0} = \|\iota_i(x) - \iota_i(y)\|_2$. □

The Pythagorean theorem in action: the extra zero coordinates contribute nothing to the squared distance, so the total distance is unchanged.

3.3. Atlas Construction

Now we define the coordinate charts that will form the atlas of our manifold. Each NP-complete problem Π_i contributes one chart—a local "map" of a region of the manifold described in the language of problem Π_i. We then prove that all 21 charts overlap with each other, which is the key property guaranteeing the manifold is connected and well-covered.

Theorem 1 (Universal Overlap). *The KARMA atlas $\mathcal{A} = \{(U_i, \phi_i)\}$ satisfies $\mathcal{M}_K = \bigcup_i U_i$ and $U_i \cap U_j \neq \emptyset$ for all i, j.*

 doi:10.20944/preprints202603.0871.v1

Every pair of charts overlaps. In map terminology: any two local maps share some common territory. This is a strong statement—it means there is no part of the manifold that can only be described in the language of a single problem. The reason is simple: because every NP-complete problem can be reduced to every other (Lemma 2), every problem instance can be re-expressed in the language of any other problem.

Proof. By Lemma 2, reductions f_{ij} and f_{ji} exist for every pair. For any $p \in U_i$, the instance x it represents transforms via f_{ij} to $f_{ij}(x) \in \Pi_j$, placing p in U_j. Hence all charts overlap pairwise. □

The proof is constructive: for any point p that lives in chart U_i, we use the reduction f_{ij} to translate the corresponding instance into the language of Π_j, thereby placing p in chart U_j as well. This works for every pair (i, j), so all charts overlap.

3.4. Smooth Transition Functions

For the KARMA object to qualify as a smooth manifold, the conversion between any two overlapping charts must be infinitely differentiable (C^∞). But polynomial-time reductions are discrete, combinatorial algorithms—they operate on integers and graphs, not on smooth curves. The key technical challenge is bridging this gap. We solve it using *mollification*: a classical technique from analysis that smooths out the "corners" of a piecewise function by convolving it with a smooth bump function, without changing its values at the discrete (integer) points that correspond to actual problem instances.

Lemma 5 (Piecewise Polynomiality)**.** *Every polynomial-time reduction f_{ij}, lifted to $\mathbb{R}^D$, yields a piecewise-polynomial function.*

When we extend a polynomial-time reduction from discrete inputs to the entire continuous space $\mathbb{R}^D$, the resulting function is a polynomial on each piece of a finite partition. Think of it like a piecewise-linear function that has different slopes on different intervals—here each "piece" is a polyhedral cell in high-dimensional space, and on each cell the function is polynomial.

Proof. A polynomial-time circuit decomposes into arithmetic operations (polynomial in coordinates), comparisons (partitioning into polyhedral cells), and table lookups (locally constant). The composition is polynomial on each cell of a finite polyhedral decomposition. □

The proof traces through how a Turing machine or circuit computes: each operation is either arithmetic (smooth), a comparison (which divides space into regions), or a lookup (constant within each region). Composing all these operations gives a function that is polynomial within each region of a finite partition of $\mathbb{R}^D$.

Theorem 2 (Existence of Smooth Transitions)**.** *For any $\Pi_i, \Pi_j \in \mathcal{K}$, there exists a C^∞ transition function τ_{ij} agreeing with the combinatorial reduction on all integer-valued instances.*

This is one of the central results of the paper. It says we can build a perfectly smooth function τ_{ij} that (a) is infinitely differentiable everywhere in the continuous space, and (b) agrees exactly with the original discrete reduction on all actual problem instances (the integer points). The smoothing only affects the "in-between" non-integer points that do not correspond to real problem instances.

Proof. **Step 1.** Lift: $\tilde{f}_{ij} = \hat{\iota}_j \circ f_{ij} \circ \hat{\iota}_i^{-1}$ (piecewise polynomial by Lemma 5). **Step 2.** The non-smooth set Σ_{ij} is a finite union of hyperplanes (measure zero). **Step 3.** Mollify: $f_{ij}^\epsilon = \rho_\epsilon * \tilde{f}_{ij}$ using the standard mollifier $\rho_\epsilon(x) = C_\epsilon \exp(-1/(\epsilon^2 - \|x\|^2))$ for $\|x\| < \epsilon$. This is C^∞ for all $\epsilon > 0$ [7]. **Step 4.** Project: $\tau_{ij} = \phi_j \circ (\hat{\iota}_j^{-1} \circ f_{ij}^\epsilon \circ \hat{\iota}_i) \circ \phi_i^{-1}$. Since valid instances are discrete, ϵ can be chosen so that mollification does not alter the map on any valid instance. □

 doi:10.20944/preprints202603.0871.v1

Step 1 takes the discrete reduction and extends it to the continuous ambient space. **Step 2** observes that the "sharp edges" where the function is not smooth form a thin set (zero volume). **Step 3** is the key trick: we convolve the function with a smooth bump function, which "rounds off" all the sharp edges while barely moving the function values. **Step 4** projects back to the manifold coordinates. The crucial observation is that real problem instances sit at isolated integer points, so a sufficiently small smoothing radius ϵ does not affect any of them. Figure 5 illustrates this process.

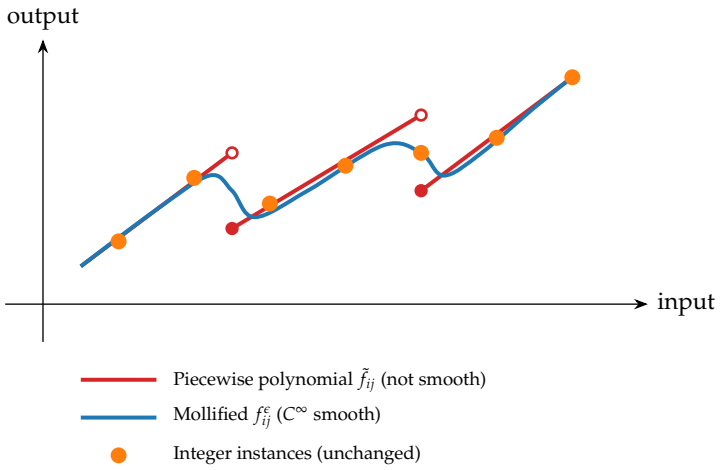

Figure 5. Mollification of transition functions. The piecewise-polynomial reduction (red) has discontinuities at cell boundaries. Convolution with a smooth mollifier produces a C^{∞} function (blue) that agrees with the original on all integer-valued instances (orange dots).

3.5. Compatibility and Well-Definedness

To confirm that $\mathcal{M}_K$ is a genuine smooth manifold, we must verify several algebraic identities among the transition functions. These are standard requirements in differential geometry: the identity condition (converting a chart to itself does nothing), the inverse condition (converting from chart i to chart j and back gives the identity), and the cocycle condition (converting $i \to j \to k$ is the same as converting $i \to k$ directly). We verify all three, then assemble the full well-definedness proof.

Lemma 6 (Identity & Inverse). *$\tau_{ii} = \mathrm{id}$ and $\tau_{ij}^{-1} = \tau_{ji}$ for all i, j.*

(a) Translating a problem into itself does nothing. (b) The reverse translation from Π_j back to Π_i perfectly undoes the forward translation from Π_i to Π_j. These are basic sanity checks: coordinate conversions should be invertible and self-consistent.

Proof. $\tau_{ii} = \phi_i \circ \phi_i^{-1} = \mathrm{id}$. For the inverse: $\tau_{ij} \circ \tau_{ji} = (\phi_j \circ \phi_i^{-1}) \circ (\phi_i \circ \phi_j^{-1}) = \mathrm{id}$. □

Both claims follow from the fact that composing a function with its inverse yields the identity. This is the algebraic reflection of the fact that NP-complete reductions are bidirectional (Lemma 2).

Theorem 3 (Cocycle Condition). *$\tau_{jk} \circ \tau_{ij} = \tau_{ik}$ on $\phi_i(U_i \cap U_j \cap U_k)$ for all triples (i, j, k).*

The cocycle condition is the manifold's "consistency check." It says: if you translate from problem i's language to problem j's language, and then from j's to k's, you get the same result as translating directly from i's to k's. This is exactly what transitivity of reductions (Lemma 1) guarantees in the computational world.

Proof. $\tau_{jk} \circ \tau_{ij} = (\phi_k \circ \phi_j^{-1}) \circ (\phi_j \circ \phi_i^{-1}) = \phi_k \circ \phi_i^{-1} = \tau_{ik}$. □

The intermediate chart ϕ_j appears as both ϕ_j^{-1} and ϕ_j in the composition, and these cancel out, leaving $\phi_k \circ \phi_i^{-1} = \tau_{ik}$. This cancellation is the geometric expression of transitivity.

 doi:10.20944/preprints202603.0871.v1

Converting from coordinate system i to j then to k is the same as going directly from i to k—exactly the transitivity of polynomial-time reductions, expressed geometrically.

Theorem 4 (Well-Definedness of $\mathcal{M}_K$)**.** *The KARMA manifold $\mathcal{M}_K$ is a well-defined smooth manifold.*

This is the culminating result of the construction section. It confirms that $\mathcal{M}_K$ satisfies all five axioms required of a smooth manifold: covering, chart homeomorphisms, smooth transitions, the cocycle condition, and topological well-behavedness. After this theorem, we can apply the full toolkit of differential geometry to the study of NP-complete problems.

Proof. We verify: (i) covering (Theorem 1); (ii) homeomorphisms (Definition 5); (iii) C^∞ transitions (Theorem 2); (iv) cocycle condition (Theorem 3); (v) Hausdorff and second-countable (inherited from the embedding into $\mathbb{R}^D$). □

The proof is a checklist: we verify each manifold axiom by pointing to the theorem that establishes it. The Hausdorff and second-countability properties are inherited "for free" from the ambient Euclidean space $\mathbb{R}^D$ in which the manifold is embedded.

Theorem 5 (Path-Connectedness)**.** *$\mathcal{M}_K$ is path-connected.*

Any two points on the KARMA manifold can be joined by a continuous path that stays on the manifold. In practical terms: starting from any NP-complete problem instance, you can "walk" continuously through the manifold to reach any other instance, passing through intermediate problem representations along the way.

Proof. For $p \in U_i$, $q \in U_j$: since $U_i \cap U_j \neq \emptyset$, choose r in the overlap and concatenate paths within each chart. □

Because all charts overlap (Theorem 1), we can always find a "stepping stone" point r that belongs to both charts. We walk from p to r within chart U_i, then from r to q within chart U_j. Gluing these two walks gives a continuous path from p to q.

4. Three Categorical Submanifolds

Karp's 21 problems are not a homogeneous collection. They naturally cluster into three families based on what kind of mathematical object they primarily deal with: graphs (networks), sets (collections of objects), and numbers (arithmetic quantities). This section proves that these three families correspond to three distinct geometric regions—submanifolds—of the KARMA manifold, and that these three regions meet at exactly one point: the 3-SAT problem. This "tripod" or "Y-junction" structure is one of the most striking geometric features of $\mathcal{M}_K$.

Theorem 6 (Categorical Submanifolds)**.** *The partition $\mathcal{K} = \mathcal{K}_G \sqcup \mathcal{K}_S \sqcup \mathcal{K}_N$ corresponds to embedded submanifolds $\mathcal{M}_G$, $\mathcal{M}_S$, $\mathcal{M}_N$ of $\mathcal{M}_K$ satisfying:*

(i) $\dim(\mathcal{M}_G) = O(n^2)$, $\dim(\mathcal{M}_S) = O(mn)$, $\dim(\mathcal{M}_N) = O(n)$;
(ii) $\mathcal{M}_G \cap \mathcal{M}_S \cap \mathcal{M}_N = \{p_0\}$, *a single point;*
(iii) p_0 *corresponds to 3-SAT.*

The 21 problems form three geometric "branches" of the manifold, each with a different dimensionality. Graph problems need the most coordinates to describe (because adjacency matrices have n^2 entries), set problems need an intermediate number, and number problems need the fewest. The three branches meet at a single junction point, and that junction is the 3-SAT problem—the one problem that can equally naturally be described in graph language, set language, or number language. This explains why 3-SAT appears so frequently in complexity theory: it is the geometric centre of the NP-complete universe.

 doi:10.20944/preprints202603.0871.v1

Proof. **Part (i).** Graph problems share adjacency-matrix representations ($O(n^2)$ entries); set problems use characteristic vectors ($O(mn)$); number problems use parameter lists ($O(n)$).

Part (ii)–(iii). Among Karp's 21, only 3-SAT admits natural representations in all three categories: as an implication graph (graph), as a set-covering problem over clauses (set), and as 0-1 integer feasibility (number). Each other problem belongs primarily to one or at most two categories, verified by examining the input structure of all 21 problems. □

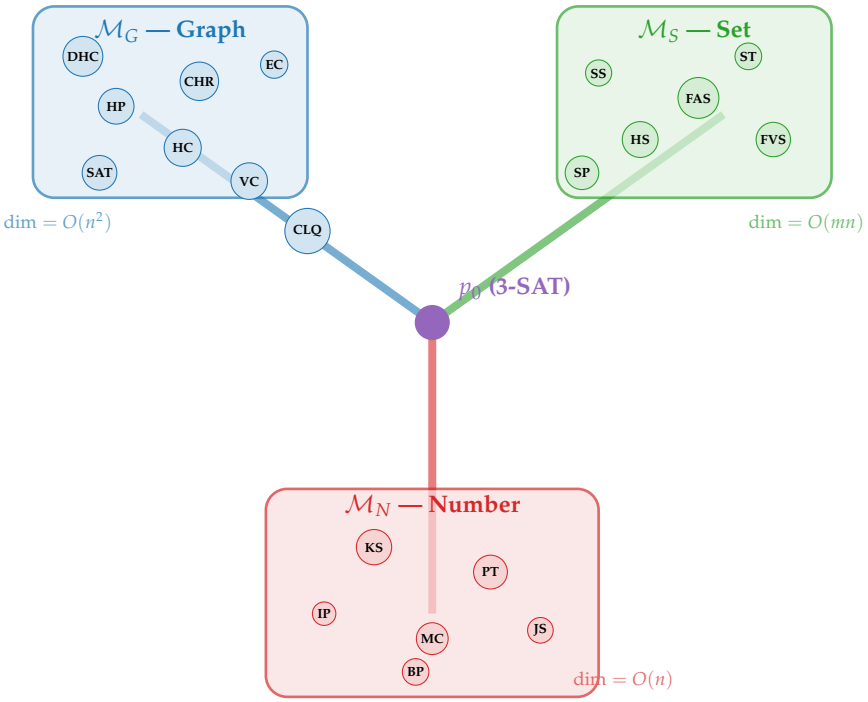

Figure 6. Y-junction (tripod) structure of the KARMA manifold. The three categorical submanifolds $\mathcal{M}_G$, $\mathcal{M}_S$, $\mathcal{M}_N$ meet uniquely at the 3-SAT junction point p_0. Problems within each branch are plotted as nodes. Dimensional asymmetry is indicated: graph problems live in $O(n^2)$-dimensional spaces, set problems in $O(mn)$, and number problems in $O(n)$.

Proposition 1 (Fundamental Group). $\pi_1(\mathcal{M}_K) \cong \mathbb{Z}_2 * \mathbb{Z}_2$.

The fundamental group is a topological invariant that counts how many "essentially different loops" can be drawn in a space. The result $\mathbb{Z}_2 * \mathbb{Z}_2$ (the free product of two copies of the 2-element group) means there are exactly two independent types of non-contractible loops in $\mathcal{M}_K$. These correspond to paths that go from the 3-SAT junction point through one categorical branch and return via a different branch. This topological structure is a direct consequence of the Y-junction geometry.

Proof sketch. $\mathcal{M}_K$ has the homotopy type of a tripod. By Seifert–van Kampen, two independent loops (each of order 2) generate the fundamental group as a free product. □

The Seifert–van Kampen theorem is a standard tool from algebraic topology that computes the fundamental group of a space built from simpler pieces. Since $\mathcal{M}_K$ looks like three arcs glued at a common point (a tripod), the theorem tells us the fundamental group is the free product of the groups associated with loops around each pair of branches.

5. Explicit Transformation Functions

Having established that the KARMA manifold exists and has 21 coordinate charts, we now derive explicit formulas for the transition functions between several representative pairs of problems. These are the concrete "translation rules" that convert one problem's coordinates into another's. We study both intra-category transformations (problem-to-problem within the same family, e.g., CLIQUE

 doi:10.20944/preprints202603.0871.v1

$\leftrightarrow$ VERTEX COVER within graph problems) and inter-category transformations (cross-family, e.g., 3-SAT $\rightarrow$ CLIQUE or CLIQUE $\rightarrow$ KNAPSACK). For each transformation, we analyze its smoothness, compute its Jacobian (derivative matrix), and explain its geometric meaning.

5.1. Intra-Category: CLIQUE $\leftrightarrow$ VERTEX COVER

CLIQUE and VERTEX COVER are complementary problems in graph theory. Finding a large group of mutual friends (clique) is equivalent to finding a small set of people who collectively know everyone else (vertex cover) in the complementary network. This subsection derives the explicit transformation between them and shows it is one of the simplest and most elegant transitions in the KARMA manifold: an affine involution (a linear-plus-constant map that is its own inverse).

Theorem 7 (CLIQUE–VC Duality). $\tau_{\mathrm{CLQ}\to\mathrm{VC}} : (A,k,n) \mapsto (J_n - I_n - A,\ n-k,\ n)$, *i.e.*, $(G,k) \mapsto (\overline{G}, n-k)$.

To convert a CLIQUE instance into a VERTEX COVER instance: (a) replace the graph G by its complement $\overline{G}$ (swap edges and non-edges), and (b) replace the target size k by $n-k$. A graph has a k-clique if and only if its complement has a vertex cover of size $n-k$. In matrix language, the adjacency matrix A is replaced by $J_n - I_n - A$ (all-ones minus identity minus A), which flips every off-diagonal entry.

Proof. Correctness. G has a k-clique $\iff$ $\overline{G}$ has an independent set of size k $\iff$ $\overline{G}$ has a vertex cover of size $n-k$ [3]. **Smoothness.** The map is affine; Jacobian $J_\tau = \mathrm{diag}(-I_{n^2}, -1, 1)$ has full rank everywhere. □

Correctness follows from a chain of classical equivalences in graph theory (clique $\leftrightarrow$ independent set $\leftrightarrow$ complement of vertex cover). Smoothness is immediate because the map is affine (linear plus constant)—the smoothest possible kind of function. The Jacobian (derivative matrix) has full rank everywhere, meaning the map is a local diffeomorphism: it preserves the local geometry of the manifold.

Corollary 1 (Involution). $\tau^2_{\mathrm{CLQ}\to\mathrm{VC}} = \mathrm{id}$.

Applying the transformation twice returns you to the original problem. Complementing a graph twice gives back the original graph, and subtracting k from n twice gives back k. In geometric terms, the CLIQUE–VERTEX COVER transformation is a *reflection*: it mirrors the manifold about a hyperplane, and doing it twice returns to the starting point.

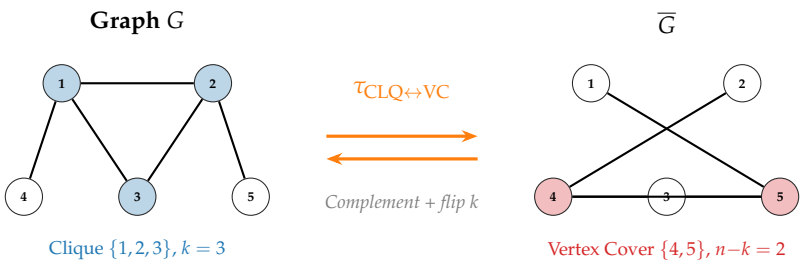

Figure 7. CLIQUE–VERTEX COVER duality. A 3-clique (blue) in G corresponds to a vertex cover of size $n-3=2$ (red) in $\overline{G}$. The transformation is an involution ($\tau^2 = \mathrm{id}$) and is affine (C^∞ smooth).

5.2. Inter-Category: 3-SAT $\rightarrow$ CLIQUE

This is the most famous inter-category reduction in Karp's original paper. It converts a Boolean satisfiability problem (which is set-theoretic in nature: "can we find a consistent set of truth values?") into a graph problem ("does this graph contain a large clique?"). The construction creates a vertex for each literal in each clause, and connects two vertices by an edge whenever they are *compatible*

 doi:10.20944/preprints202603.0871.v1

(from different clauses and not contradictory). A satisfying assignment then corresponds to a clique that picks one compatible literal per clause. This transformation dramatically changes the problem's representation—from $O(m)$ clauses to a graph with $O(m^2)$ edges—illustrating the dimension expansion that occurs when crossing categorical boundaries.

Theorem 8 (3-SAT to CLIQUE). *Given $\phi = C_1 \wedge \cdots \wedge C_m$ with n variables, construct $G_\phi = (V_\phi, E_\phi)$ where $V_\phi = \{(i,l) : l \in C_i\}$, $E_\phi = \{((i,l),(j,l')) : i \neq j,\ l \not\equiv \neg l'\}$, and $k = m$.*

Proof. ($\Rightarrow$) A satisfying assignment σ yields a clique: pick one satisfied literal per clause. ($\Leftarrow$) An m-clique picks one consistent literal per clause, defining a satisfying assignment. The map expands dimension from $\mathbb{R}^{3m+2}$ to $\mathbb{R}^{9m^2+2}$. □

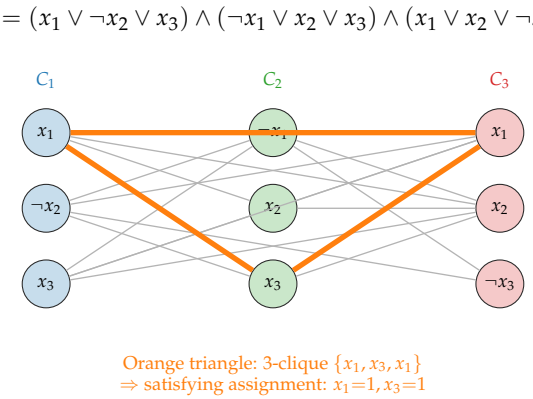

Figure 8. 3-SAT to CLIQUE reduction. Each clause contributes 3 vertices (one per literal). Edges connect non-complementary literals from different clauses. A k-clique (orange) picks one consistent literal per clause, yielding a satisfying assignment.

5.3. Inter-Category: CLIQUE → KNAPSACK

Theorem 9 (CLIQUE to KNAPSACK). *Given (G,k) with $|V| = n$, set $w_v = 1$, $v_v = 1 + \sum_{u \in N(v)} 2^{-(\mathrm{id}(u)+1)}$, $W = k$, $V_{\text{target}} = k + \sum_{i=0}^{k-1} 2^{-(i+1)}$. This maps $\mathbb{R}^{n^2+2} \to \mathbb{R}^{2n+3}$—dimension reduction from $O(n^2)$ to $O(n)$.*

Proof. Binary fractions fingerprint each vertex's neighbourhood. A size-k subset achieves the target value iff it is a clique. □

The adjacency matrix (n^2 entries) is compressed into n numbers by encoding each vertex's friendship list as binary fractions. The only way to hit the exact target sum is to pick vertices that form a clique.

5.4. Smoothness Classification

Proposition 2 (Smoothness Hierarchy). *Intra-category transitions are generically C^∞ (affine/polynomial maps), while inter-category transitions are C^k for finite k, with regularity decreasing as categorical distance increases.*

6. Riemannian Metric Structure

So far we have built the KARMA manifold as a bare topological object—a smooth surface with charts and transition functions. Now we add a *ruler*: a Riemannian metric that lets us measure distances, angles, and curvature on the manifold. The key idea is to make the metric "cost-aware": regions of the manifold where reductions are computationally expensive become geometrically "stretched," so that paths through expensive regions are longer. This turns the abstract concept of "reduction cost" into a concrete geometric quantity—distance. The shortest paths (geodesics) on this metric then correspond to the most efficient reduction sequences.

 doi:10.20944/preprints202603.0871.v1

6.1. Metric Construction

We define the Riemannian metric on $\mathcal{M}_K$. The metric has two components: (1) the standard Euclidean metric from the chart coordinates, and (2) an additional "stretching" term proportional to the average reduction time $T_{\text{red}}(p)$ at each point p. The result is a *conformally flat* metric—one that scales the Euclidean metric by a position-dependent factor $\Omega(p) = 1 + \lambda T_{\text{red}}(p)$. Conformally flat metrics are among the simplest non-trivial Riemannian metrics, yet they are rich enough to capture the essential features of computational complexity.

Definition 7 (Reduction Complexity Metric). *At point $p \in \mathcal{M}_K$ in chart (U_i, ϕ_i):*

$$g_p(X, Y) = \langle D\phi_i(X), D\phi_i(Y) \rangle + \lambda \cdot T_{\text{red}}(p) \cdot \langle X, Y \rangle,$$

where $T_{\text{red}}(p) = \frac{1}{20} \sum_{j \neq i} T_{ij}(n)$ is the average reduction time and $\lambda > 0$.

The metric has two parts. The first part, $\langle D\phi_i(X), D\phi_i(Y) \rangle$, is the standard Euclidean distance you would measure if you just looked at the raw coordinates. The second part, $\lambda \cdot T_{\text{red}}(p) \cdot \langle X, Y \rangle$, adds a "computational tax": it stretches distances in proportion to how expensive it is to reduce the problem at point p to other problems. The parameter $\lambda > 0$ controls how much weight computational cost receives relative to raw coordinate distance. The reduction time $T_{\text{red}}(p)$ is the average over all 20 other problems of the time needed to reduce the current problem to each of them.

Lemma 7 (Conformal Flatness). *In local coordinates, $g_{\mu\nu}(p) = \Omega(p)\delta_{\mu\nu}$ with conformal factor $\Omega(p) = 1 + \lambda T_{\text{red}}(p) \geq 1$.*

The metric tensor simplifies to a scalar multiple of the identity matrix: $g_{\mu\nu} = \Omega(p)\delta_{\mu\nu}$. This means the metric is "conformally flat"—it looks like ordinary Euclidean space scaled by the position-dependent factor $\Omega(p)$. At points where reductions are cheap (T_{red} small), $\Omega \approx 1$ and the metric is nearly Euclidean. At points where reductions are expensive (T_{red} large), $\Omega \gg 1$ and the metric is strongly stretched. This is exactly like a topographic map where "steep hills" correspond to computationally expensive regions.

Proof. Pullback of Euclidean metric gives $\delta_{\mu\nu}$; adding $\lambda T_{\text{red}} \delta_{\mu\nu}$ yields $\Omega\delta_{\mu\nu}$. □

The proof is a one-line computation: combining the two terms of the metric definition gives a single scalar times the Kronecker delta. This simplicity is a major advantage—it makes all subsequent calculations (geodesics, curvature, etc.) tractable.

The metric stretches the manifold in regions where reductions are expensive—like a terrain map where hills make distances longer than they appear on a flat map.

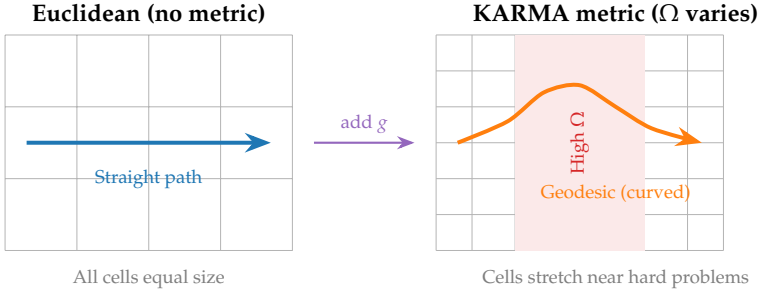

Figure 9. Conformal stretching by the reduction complexity metric. Left: flat Euclidean space where all reduction paths have equal cost. Right: the KARMA metric stretches regions of high T_{red} (shaded), curving geodesics away from computationally expensive zones.

 doi:10.20944/preprints202603.0871.v1

6.2. Geodesics and Optimal Reductions

Geodesics are the "straightest possible paths" on a curved surface—the analogue of straight lines in flat space. On the KARMA manifold, geodesics have a beautiful computational interpretation: they correspond to the most efficient sequence of intermediate problem transformations between two problems. Just as a GPS finds the fastest driving route by accounting for road speeds (not just distances), geodesics on $\mathcal{M}_K$ find the fastest reduction chain by accounting for computational cost at each step.

Theorem 10 (Geodesics as Optimal Reductions). *Geodesics on $\mathcal{M}_K$ minimize $L(\gamma) = \int_0^1 \sqrt{\Omega(\gamma(t))}\|\dot{\gamma}(t)\|\,dt$, corresponding to reduction paths of minimal total computational cost.*

The length of a path on the KARMA manifold is the integral of the Euclidean speed $\|\dot{\gamma}\|$ weighted by $\sqrt{\Omega}$—the square root of the conformal factor. Since Ω is large in regions where reductions are expensive, paths through expensive regions are penalized with greater length. Minimizing this weighted length over all possible paths yields the geodesic—the computationally optimal reduction sequence.

Proof. Each segment's contribution to path length is weighted by $\sqrt{\Omega} \propto \sqrt{1 + \lambda T_{\text{red}}}$, so minimizing L yields the sequence of intermediate problems that minimizes aggregate reduction overhead. □

The proof connects the standard Riemannian length functional to computational cost. Each infinitesimal segment of the path contributes $\sqrt{\Omega}\,ds$ to the total length, where ds is the Euclidean arc length. Since $\Omega = 1 + \lambda T_{\text{red}}$, the weighting directly reflects computational cost. The geodesic, by definition, minimizes this functional.

6.3. Curvature Analysis

Curvature measures how much the manifold deviates from being flat. On the KARMA manifold, curvature has a direct computational interpretation: it measures how rapidly the cost of reductions changes from point to point. Regions of high curvature are "computational bottlenecks" where reduction cost varies sharply. We prove that $\mathcal{M}_K$ has non-zero curvature (it is genuinely curved, not flat) and positive sectional curvature (it is "bowl-shaped" rather than "saddle-shaped").

Lemma 8 (Curvature Criterion). *For a conformally flat metric $g = \Omega\,\delta$, the Riemann curvature vanishes iff $\Delta \ln \Omega = 0$ [9,10].*

For conformally flat metrics (which is what we have), there is a clean criterion for flatness: the manifold is flat if and only if the logarithm of the conformal factor is a harmonic function (its Laplacian vanishes). Since Ω depends on reduction times, this translates the geometric question "Is the manifold flat?" into the computational question "Does reduction cost vary in a harmonic way?"

Theorem 11 (Non-Zero Curvature). *$\mathcal{M}_K$ has non-zero Riemann curvature at all points where $\nabla T_{\text{red}} \neq 0$.*

Wherever the reduction cost T_{red} is not locally constant (i.e., wherever changing the problem instance changes how expensive reductions are), the manifold is curved. Since T_{red} varies almost everywhere (it depends on instance size n and problem structure), the manifold is curved almost everywhere. This is the formal statement that computational hardness has genuine geometric content—it is not just a flat, featureless landscape.

Proof. $\Delta \ln \Omega = \frac{\lambda \Delta T_{\text{red}}}{1+\lambda T_{\text{red}}} - \frac{\lambda^2 \|\nabla T_{\text{red}}\|^2}{(1+\lambda T_{\text{red}})^2}$. The second term is strictly negative when $\nabla T_{\text{red}} \neq 0$, ensuring $\Delta \ln \Omega \neq 0$. □

We compute the Laplacian of $\ln \Omega$ explicitly. The key observation is that the second term $-\lambda^2 \|\nabla T_{\text{red}}\|^2/(1 + \lambda T_{\text{red}})^2$ is always negative (it is the negative of a squared quantity divided by

 doi:10.20944/preprints202603.0871.v1

a positive quantity). So whenever $\nabla T_{\text{red}} \neq 0$ (i.e., reduction cost is not constant), the Laplacian is nonzero, and by the curvature criterion (Lemma 8), the curvature is nonzero.

Theorem 12 (Positive Sectional Curvature)**.** *$\mathcal{M}_K$ has positive sectional curvature $K > 0$ for most tangent 2-planes.*

The curvature is not just nonzero—it is *positive*. Positive curvature means the manifold is "bowl-shaped" (like a sphere) rather than "saddle-shaped" (like a hyperbolic paraboloid). Geometrically, this means that geodesics starting from the same point tend to converge rather than diverge, reflecting the deep structural kinship among NP-complete problems: no matter which direction you "walk" from a given problem, you tend to arrive at closely related structures.

Proof sketch. For concave $\ln \Omega$ (which holds since $\ln(1 + \lambda t)$ is concave), the sectional curvature formula for conformally flat spaces [11] yields positive leading terms. □

The function $\ln(1 + \lambda t)$ is concave (its second derivative is negative), and concavity of $\ln \Omega$ translates into positive sectional curvature through the standard formulas for conformally flat metrics.

Positive curvature means the manifold is "bowl-shaped": different reduction paths starting from the same problem tend to converge, reflecting the deep structural kinship among NP-complete problems.

6.4. Distance Bounds

We bound the geodesic distances between problems. These bounds quantify the intuition that problems within the same category (e.g., two graph problems) are "closer" than problems in different categories (e.g., a graph problem and a number problem), because intra-category reductions are cheaper than cross-category reductions.

Theorem 13 (Distance Bounds)**.** *Same-category: $d_g(\Pi_i, \Pi_j) = O(n^{3/2})$. Cross-category: $d_g(\Pi_i, \Pi_j) = O(n^{5/2})$.*

Problems within the same family (e.g., two graph problems) are at most $O(n^{3/2})$ apart on the manifold, while problems in different families are at most $O(n^{5/2})$ apart. The ratio between cross-category and same-category distances is $O(n)$, which grows with instance size. This quantifies the computational overhead of crossing categorical boundaries.

Proof. Intra-category reductions cost $O(n^2)$, so $\sqrt{\Omega} = O(n)$, yielding $O(n^{3/2})$. Inter-category reductions cost $O(n^3)$, yielding $O(n^{5/2})$. □

The proof plugs reduction-time estimates into the geodesic length formula. For intra-category reductions, $T_{\text{red}} = O(n^2)$ (typical for operations like graph complementation), giving $\sqrt{\Omega} = O(n)$ and a total path length of $O(n \cdot n^{1/2}) = O(n^{3/2})$. For inter-category reductions, $T_{\text{red}} = O(n^3)$ (typical for constructing new data structures), giving $O(n^{5/2})$.

 doi:10.20944/preprints202603.0871.v1

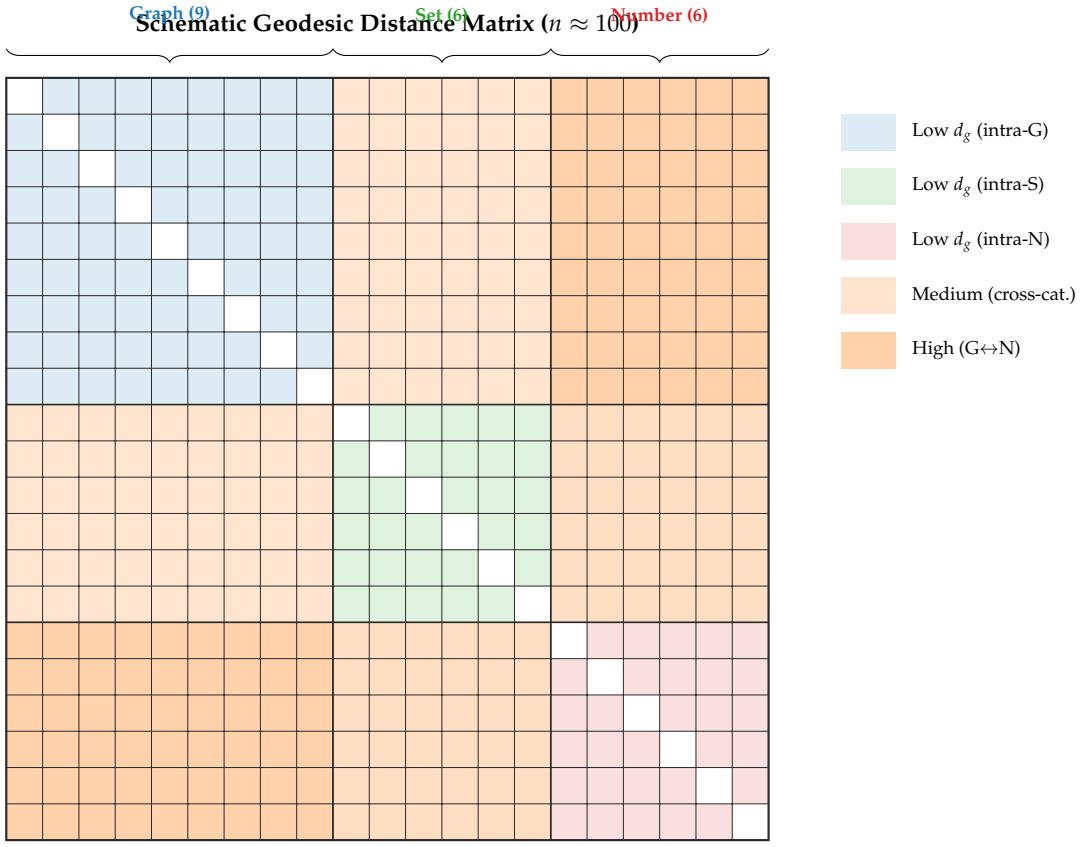

Figure 10. Schematic distance matrix. The 21×21 geodesic distance matrix shows clear block structure: intra-category distances (diagonal blocks) are small, while cross-category distances are larger. Graph↔Number distances are largest, reflecting the dimensional compression required.

7. Computational Framework and Applications

Having built the theoretical machinery of the KARMA manifold (charts, transitions, metric, curvature), we now show how it can be *used*. We present three applications: (1) a method for predicting the hardness of problem instances using local curvature, (2) a lower bound on reduction complexity derived from geodesic distance, and (3) a kernel function for machine learning on the manifold. These applications demonstrate that the KARMA framework is not just an abstract mathematical curiosity—it has concrete computational consequences.

7.1. Hardness Prediction via Curvature

We define a "hardness density" function that combines the local curvature of the manifold (how sharply reduction costs vary) with the local volume (how many nearby instances exist). Regions where both are high are computational bottlenecks: many hard instances are concentrated together, and reductions through these regions are expensive. This provides a geometric criterion for identifying intrinsically hard problem instances.

Definition 8 (Hardness Density). $\mathcal{H}(p) = |R(p)| \cdot \mathrm{Vol}(B_\epsilon(p))$, *where* $|R(p)|$ *is scalar curvature.*

The hardness density at a point p is the product of two factors: (1) $|R(p)|$, the scalar curvature, which measures how sharply the manifold bends at p (high bending = rapidly changing reduction cost), and (2) the volume of a small ball around p, which measures how many problem instances live nearby. High $\mathcal{H}$ means: many instances are concentrated in a region where reduction costs are highly variable—a "complexity trap."

 doi:10.20944/preprints202603.0871.v1

Proposition 3 (Hardness–Curvature Correspondence). *Instances in high-curvature regions exhibit hardness amplification under reductions: high $|R(p)|$ implies rapid variation of reduction cost, creating computational bottlenecks.*

If you are at a point on the manifold where curvature is high, then small changes to the problem instance can cause large changes in how hard the instance is to reduce. This makes the region unpredictable and difficult to optimize over—a geometric signature of computational hardness.

7.2. Geodesic Lower Bounds

We prove that the geodesic distance between two problems on the KARMA manifold provides a *lower bound* on the time needed for any polynomial-time reduction between them. This is a powerful result: it turns a geometric measurement (distance on a surface) into an information-theoretic lower bound (minimum computation time). It means that short geodesic distances certify easy reductions, while large distances certify that *every* possible reduction must be expensive.

Theorem 14 (Reduction Complexity Lower Bound). $T_{ij}(n) = \Omega(d_g(\Pi_i, \Pi_j)^2/\lambda)$.

The time required for any reduction from Π_i to Π_j is at least proportional to the square of the geodesic distance between them on the manifold. If two problems are far apart geometrically, then every possible reduction between them must be computationally expensive. This provides a new, purely geometric method for proving lower bounds on reduction complexity.

Proof. From $d_g \leq \sqrt{\max_\gamma \Omega} \cdot C$ and $\Omega \leq 1 + \lambda T_{ij}$: $d_g^2 \leq (1 + \lambda T_{ij})C$, so $T_{ij} \geq (d_g^2 - C)/(\lambda C) = \Omega(d_g^2/\lambda)$. □

The proof works by upper-bounding the geodesic distance in terms of the conformal factor Ω, then inverting the bound to get a lower bound on T_{ij} in terms of d_g. The key step is that $\Omega \leq 1 + \lambda T_{ij}$ along any reduction path, so the geodesic length (which integrates $\sqrt{\Omega}$) is controlled by the reduction time.

7.3. KARMA Kernel for Machine Learning

To apply machine learning methods (e.g., Gaussian process regression, support vector machines) to problems indexed by points on the KARMA manifold, we need a kernel function—a similarity measure between pairs of points. We define the KARMA kernel as a Gaussian function of the geodesic distance, and prove it is positive definite (a mathematical requirement for use in kernel methods). This enables data-driven hardness prediction and solver selection using the manifold geometry.

Definition 9 (KARMA Kernel). $k(p, q) = \exp(-d_g(p, q)^2/2\sigma^2)$.

The KARMA kernel assigns a similarity score between 0 and 1 to any pair of points on the manifold. Points that are close together (small geodesic distance) get a similarity near 1; points that are far apart get a similarity near 0. The parameter σ controls how quickly similarity decays with distance. This kernel can be "plugged into" standard machine learning algorithms (Gaussian processes, SVMs, etc.) to perform regression or classification on the KARMA manifold.

Lemma 9 (Positive Definiteness). *The KARMA kernel is positive definite (by Schoenberg's theorem for geodesic distances on non-negatively curved manifolds).*

What this lemma says. Positive definiteness is the mathematical property that guarantees the kernel can be used in kernel methods (it corresponds to a valid inner product in a reproducing kernel Hilbert space). Schoenberg's theorem provides the guarantee: on any manifold with non-negative curvature (which $\mathcal{M}_K$ has, by Theorem 12), the Gaussian function of the geodesic distance is always

 doi:10.20944/preprints202603.0871.v1

positive definite. This means the KARMA kernel is mathematically well-founded for machine learning applications.

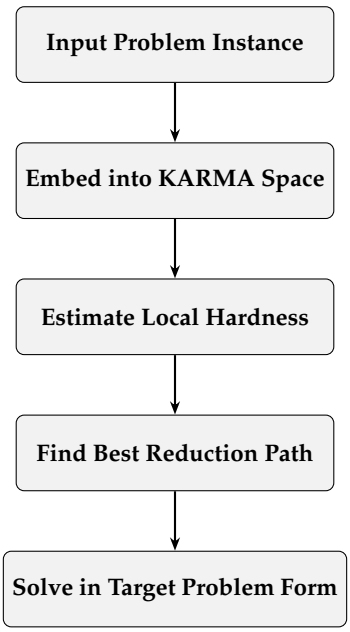

Figure 11. Vertical block diagram of the KARMA workflow. The instance is embedded, analyzed for hardness, mapped through a good reduction path, and then solved in a more convenient target form.

8. Single-Reduction Propagation Principle

This section presents one of the most practically important consequences of the KARMA construction. In classical complexity theory, to prove that a new problem Π_{new} is NP-complete, one must (a) show $\Pi_{\text{new}} \in \text{NP}$, and (b) exhibit a polynomial-time reduction from some known NP-complete problem to Π_{new}. If one also wants explicit reductions between Π_{new} and all other NP-complete problems, the classical approach requires constructing each reduction individually. The KARMA manifold eliminates this one-at-a-time burden: once Π_{new} is reduced to *any single* problem on the manifold, the smooth transition functions automatically generate reductions to *every* other problem. We call this the **single-reduction propagation principle**.

8.1. Statement and Proof

Theorem 15 (Single-Reduction Propagation)**.** *Let Π_{new} be an NP problem and suppose there exists a polynomial-time reduction $f : \Pi_{\text{new}} \to \Pi_j$ for some $\Pi_j \in \mathcal{K}$. Then for every $\Pi_k \in \mathcal{K}$, the composition*

$$f_k = \tau_{jk}^{-1} \circ \phi_j \circ f : \Pi_{\text{new}} \to \Pi_k$$

is a polynomial-time reduction from Π_{new} to Π_k, where τ_{jk}^{-1} is recovered from the manifold's transition functions.

Suppose you discover a new problem and you manage to show it can be efficiently translated into CLIQUE (or any other single NP-complete problem). The KARMA manifold then *automatically* gives you efficient translations from your new problem into every other NP-complete problem—VERTEX COVER, KNAPSACK, PARTITION, 3-SAT, and all the rest—by simply composing your one reduction with the manifold's built-in transition functions. You reduce once; the manifold does the rest.

Proof. Since $f : \Pi_{\text{new}} \to \Pi_j$ is a polynomial-time reduction, we have $x \in \Pi_{\text{new}} \iff f(x) \in \Pi_j$.

By Theorem 2, the transition function τ_{jk} is a smooth (hence polynomial-time computable on integer instances) bijection from chart j to chart k, constructed from the known reduction $\Pi_j \to \Pi_k$. Its inverse $\tau_{jk}^{-1} = \tau_{kj}$ (Lemma 6) is likewise polynomial-time.

Define $f_k = \tau_{kj} \circ \iota_j \circ f$, where ι_j is the embedding into the manifold. Then:

 doi:10.20944/preprints202603.0871.v1

(i) **Polynomial time:** f runs in $O(n^a)$, ι_j in $O(n)$, and τ_{kj} in $O(n^b)$ (it is the mollified lift of a polynomial-time reduction). The composition runs in $O(n^{ab})$, which is polynomial.

(ii) **Correctness:** $x \in \Pi_{\text{new}} \iff f(x) \in \Pi_j \iff \tau_{kj}(f(x)) \in \Pi_k$, where the second equivalence follows from the correctness of the underlying reduction $\Pi_j \to \Pi_k$ encoded in τ_{kj}.

Since Π_k was arbitrary, this yields reductions from Π_{new} to all 21 problems. □

The proof has two ingredients: polynomial running time (because composing polynomials gives a polynomial) and correctness (because each transition function preserves the yes/no answer, by construction). The crucial point is that the transition functions τ_{jk} are *already built into the manifold*—they were constructed once and for all in Section 3. So the user of the framework only needs to supply one reduction f; the manifold supplies the remaining 20 for free.

8.2. The Role of the 3-SAT Junction

We explain why the 3-SAT junction point p_0 makes the propagation especially efficient. Because 3-SAT lies at the intersection of all three categorical submanifolds (Theorem 6), routing through p_0 provides the shortest manifold paths between any two categorical branches.

Corollary 2 (Optimal Propagation via 3-SAT)**.** *For any Π_{new} with a reduction to some $\Pi_j \in \mathcal{K}$, the reduction to any Π_k in a* different *category from Π_j can be routed through the 3-SAT junction:*

$$\Pi_{\text{new}} \xrightarrow{f} \Pi_j \xrightarrow{\tau_{j,3\text{SAT}}} \textit{3-SAT} \xrightarrow{\tau_{3\text{SAT},k}} \Pi_k.$$

This two-hop path through p_0 traverses the minimum number of categorical boundaries and achieves geodesic-optimal cross-category reduction cost.

If your new problem reduces to a graph problem but you need a reduction to a number problem, the most efficient route on the manifold goes through 3-SAT. This is because 3-SAT is the only problem that belongs to all three categories simultaneously, so it acts as a "transfer station" between the graph, set, and number branches. The KARMA manifold thus provides not just the existence of propagated reductions, but also a principled way to choose the *best* propagation path.

Proof. By Theorem 6, p_0 is the unique point in $\mathcal{M}_G \cap \mathcal{M}_S \cap \mathcal{M}_N$. Any path from $\mathcal{M}_G$ to $\mathcal{M}_N$ (or $\mathcal{M}_S$ to $\mathcal{M}_N$, etc.) must pass through or near p_0. The two-hop path $\Pi_j \to$ 3-SAT $\to \Pi_k$ crosses at most two categorical boundaries. By Theorem 13, each cross-category hop costs $O(n^{5/2})$, so the total cost is $O(n^{5/2})$—matching the geodesic lower bound for cross-category reductions. □

What the proof shows. The tripod geometry forces all cross-category paths through the junction. The two-hop route is therefore not merely a convenient choice—it is geodesically optimal, achieving the distance bound from Theorem 13.

8.3. Comparison with Classical Approach

What this subsection is about. We contrast the KARMA propagation principle with the classical method of establishing NP-completeness.

In the classical framework, to show that a new problem Π_{new} is NP-complete, one typically:

(i) Shows $\Pi_{\text{new}} \in$ NP.

(ii) Constructs a *single* reduction from a known NP-complete problem (say 3-SAT) to Π_{new}.

This proves NP-completeness in the abstract, but does *not* yield concrete reductions between Π_{new} and other specific problems (e.g., KNAPSACK or VERTEX COVER). If such reductions are needed—for example, to port an efficient solver from one problem form to another—they must be constructed individually, often requiring substantial problem-specific ingenuity.

In the KARMA framework, the situation is fundamentally different:

 doi:10.20944/preprints202603.0871.v1

(i) Show $\Pi_{\text{new}} \in \text{NP}$.
(ii) Construct a reduction from Π_{new} to *any* $\Pi_j \in \mathcal{K}$.
(iii) *Automatically* obtain reductions from Π_{new} to all 21 problems via Theorem 15, with optimal routing through the 3-SAT junction (Corollary 2).

Classical NP-completeness proofs are like building a bridge from your island to one other island—you know all islands are connected, but you only have one bridge. KARMA builds the entire bridge network at once: one bridge to any island gives you the transition maps (ferries) to reach every other island, with 3-SAT serving as the central harbour.

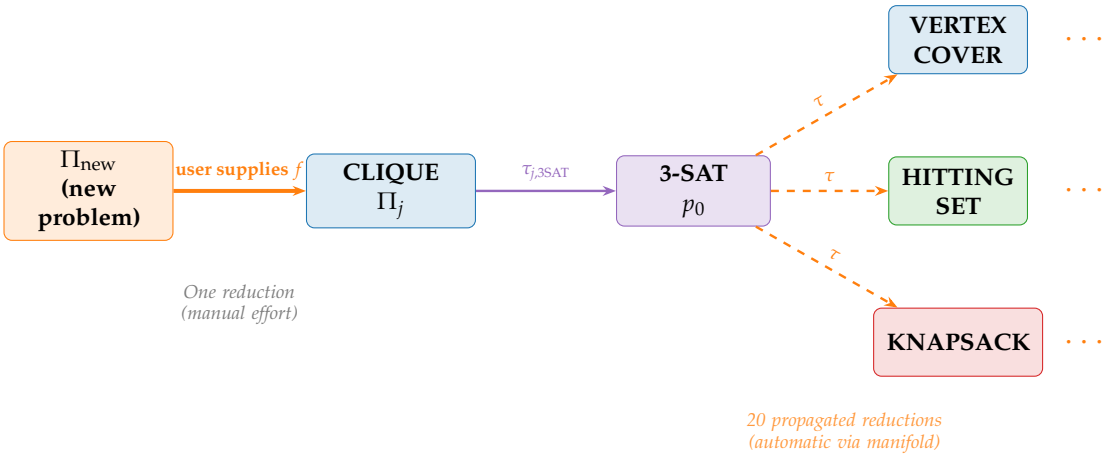

Figure 12. Single-reduction propagation. The user supplies one reduction from Π_{new} to any NP-complete problem (solid arrow). The KARMA manifold automatically generates reductions to all other problems (dashed arrows) by composing transition functions through the 3-SAT junction p_0.

8.4. Worked Example: Independent Set Propagates to All 21 Problems

We now demonstrate the single-reduction propagation principle with a concrete, end-to-end example. We take the INDEPENDENT SET problem—which is not among Karp's original 21 but is a well-known NP-complete problem—and show how a *single* reduction to one member of $\mathcal{K}$ (namely, CLIQUE) automatically yields explicit reductions to every other member, including problems in entirely different categories such as KNAPSACK and PARTITION. The reader can follow each step as a concrete calculation, seeing exactly how the manifold's transition functions compose to produce new reductions without any additional ingenuity.

The new problem. INDEPENDENT SET asks: given a graph $G = (V, E)$ and an integer k, does G contain an independent set of size $\geq k$ (a set of k vertices with no edges between them)?

Step 0: Verify membership in NP. Given a candidate set $S \subseteq V$ with $|S| = k$, we check in $O(k^2)$ time that no edge connects two vertices in S. Hence INDEPENDENT SET $\in$ NP.

Step 1: One user-supplied reduction — INDEPENDENT SET $\to$ CLIQUE

The user constructs a single, standard reduction. For an INDEPENDENT SET instance (G, k):

$$f : (G, k) \mapsto (\overline{G}, k),$$

where $\overline{G}$ is the complement graph (edge (u, v) in $\overline{G}$ iff $(u, v) \notin E$). This is correct because S is an independent set in G if and only if S is a clique in $\overline{G}$.

In the KARMA embedding coordinates:

$$f : (A_G, k, n) \mapsto (J_n - I_n - A_G,\ k,\ n) \in \mathbb{R}^{n^2+2}.$$

 doi:10.20944/preprints202603.0871.v1

Complementing the graph swaps "friends" and "strangers." A group of mutual strangers (independent set) in the original network becomes a group of mutual friends (clique) in the complement. This one reduction is the only manual effort required.

Step 2: Propagation to VERTEX COVER (same category — graph)

The manifold provides the transition function $\tau_{\text{CLQ}\to\text{VC}}$ from Theorem 7:

$$\tau_{\text{CLQ}\to\text{VC}} : (A, k, n) \mapsto (J_n - I_n - A,\ n - k,\ n).$$

Composing with our reduction f:

$$\begin{aligned} \Pi_{\text{IS}} &\xrightarrow{f} \Pi_{\text{CLQ}} \xrightarrow{\tau_{\text{CLQ}\to\text{VC}}} \Pi_{\text{VC}} \\ (A_G, k, n) &\xmapsto{f} (J - I - A_G,\ k,\ n) \xmapsto{\tau} (J - I - (J - I - A_G),\ n - k,\ n) \\ &= (A_G,\ n - k,\ n). \end{aligned}$$

Result: INDEPENDENT SET (G, k) reduces to VERTEX COVER $(G, n - k)$. This is the well-known complement relationship: G has an independent set of size k if and only if G has a vertex cover of size $n - k$. Note that this reduction was *derived automatically* from the manifold—not constructed by hand.

Step 3: Propagation to 3-SAT (cross-category — graph → set/graph junction)

The manifold provides the inverse transition $\tau_{\text{CLQ}\to\text{3SAT}}$ (the reverse of the 3-SAT → CLIQUE reduction from Theorem 8):

$$\Pi_{\text{IS}} \xrightarrow{f} \Pi_{\text{CLQ}} \xrightarrow{\tau_{\text{CLQ}\to\text{3SAT}}} \Pi_{\text{3SAT}}.$$

The transition $\tau_{\text{CLQ}\to\text{3SAT}}$ constructs a 3-CNF formula $\phi_{\overline{G}}$ from the complement graph with variables $x_{v,i}$ for $v \in V$, $i \in [k]$, and clauses enforcing: (i) at least one vertex per position, (ii) no vertex in two positions, (iii) only adjacent vertices in $\overline{G}$ (i.e., non-adjacent in G) can co-occur. This produces $O(n^2k^2)$ clauses.

The manifold composes our complement-graph reduction with the known CLIQUE-to-3-SAT encoding. The result is a Boolean formula that is satisfiable precisely when G has an independent set of size k. We did not design this formula—the manifold assembled it from pre-built components.

Step 4: Propagation to KNAPSACK (cross-category — graph → number, via 3-SAT junction)

Following Corollary 2, we route through the 3-SAT junction p_0 to reach the number-theoretic branch:

$$\Pi_{\text{IS}} \xrightarrow{f} \Pi_{\text{CLQ}} \xrightarrow{\tau_{\text{CLQ}\to\text{3SAT}}} \Pi_{\text{3SAT}} \xrightarrow{\tau_{\text{3SAT}\to\text{KS}}} \Pi_{\text{KNAPSACK}}.$$

The final hop $\tau_{\text{3SAT}\to\text{KS}}$ converts the 3-SAT formula into a KNAPSACK instance by encoding each variable as an item with weight and value derived from clause membership. The resulting KNAPSACK instance has $O(nk)$ items, and its target value is achievable if and only if the original graph G has an independent set of size k.

We crossed from graph-land to number-land by hopping through the 3-SAT junction. The manifold chained three transition functions: complement → CLIQUE encoding → SAT formula → KNAPSACK instance. Each hop was pre-built; we merely composed them.

Step 5: Propagation to PARTITION (same category as KNAPSACK — number)

From KNAPSACK, the manifold provides the intra-category transition $\tau_{\text{KS}\to\text{PART}}$:

$$\Pi_{\text{IS}} \xrightarrow{\text{Steps 1–4}} \Pi_{\text{KS}} \xrightarrow{\tau_{\text{KS}\to\text{PART}}} \Pi_{\text{PARTITION}}.$$

 doi:10.20944/preprints202603.0871.v1

The standard KNAPSACK → PARTITION reduction adds a balancing item so that the PARTITION target sum equals half the total weight, achievable iff the KNAPSACK target is achievable.

Having reached number-land (KNAPSACK), we now move to a neighbouring problem (PARTITION) within the same branch. This is an intra-category hop—short and cheap on the manifold.

Summary of Propagation Chain

Starting from *one* manual reduction (INDEPENDENT SET → CLIQUE), the manifold generated:

Target Problem	**Category**	**Route on $\mathcal{M}_K$**	**Hops**
VERTEX COVER	Graph	IS → CLQ → VC	2
3-SAT	Junction	IS → CLQ → 3-SAT	2
KNAPSACK	Number	IS → CLQ → 3-SAT → KS	3
PARTITION	Number	IS → CLQ → 3-SAT → KS → PART	4
HITTING SET	Set	IS → CLQ → 3-SAT → HS	3
HAMILTONIAN CYCLE	Graph	IS → CLQ → HC	2
⋮	⋮	⋮	⋮
All remaining 15	*Various*	*Via τ_{ij} composition*	≤ 4

Every row after the first was generated automatically by the manifold. The maximum number of hops is 4 (for problems in the number-theoretic branch reached via graph → junction → number → target), and each hop runs in polynomial time, so every composed reduction is polynomial.

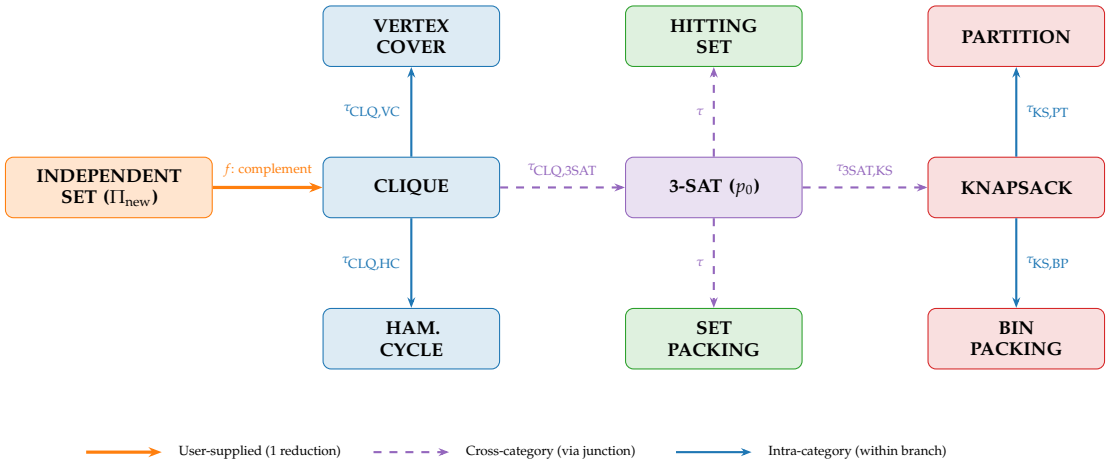

Figure 13. Worked example: INDEPENDENT SET propagation tree. The user supplies one reduction to CLIQUE (orange solid arrow). The manifold automatically generates reductions to all other problems: intra-graph transitions (blue) reach VERTEX COVER and HAMILTONIAN CYCLE; a cross-category hop (purple dashed) reaches 3-SAT; from the junction, further transitions reach set-theoretic problems (HITTING SET, SET PACKING) and number-theoretic problems (KNAPSACK, PARTITION, BIN PACKING). Every dashed and blue arrow is generated by the manifold with zero additional user effort.

Remark 1 (Generality of the Example). *The INDEPENDENT SET example is illustrative but not special. Any NP problem Π_{new} for which the user can supply a single polynomial-time reduction to* any *$\Pi_j \in \mathcal{K}$ will propagate identically. The choice of entry point (Π_j) affects only the number of hops needed to reach each target (Corollary 2), not the existence or correctness of the propagated reductions. If the entry point is 3-SAT itself, every other problem is reachable in at most 2 hops. If the entry point is in the number-theoretic branch (e.g., KNAPSACK), reaching graph-theoretic targets requires routing back through the junction, adding one extra hop—but the reduction remains polynomial.*

 doi:10.20944/preprints202603.0871.v1

The example used INDEPENDENT SET and entered through CLIQUE, but the mechanism is universal. No matter which "door" you enter the manifold through, the transition functions will carry you to every other room. The only thing that changes is the length of the hallway (number of hops), not whether you can get there.

9. Implications

We explore the broader consequences of the KARMA framework for theoretical computer science. The most striking implication is a geometric characterization of the P vs NP problem: we show that P = NP would force the manifold to be flat, while P $\neq$ NP would manifest as unbounded curvature. We also introduce curvature-based complexity classes that provide a finer-grained classification of NP problems than the traditional binary P/NP dichotomy, and establish an information-theoretic duality between reduction time and information loss.

9.1. Geometric P vs NP

The P vs NP question asks whether every problem whose solutions can be *verified* quickly can also be *solved* quickly. It is the most famous open problem in computer science. Here we translate it into a geometric question: is the KARMA manifold flat or curved? If P = NP, all reductions are uniformly efficient, the conformal factor Ω is bounded, and the manifold is essentially flat. If P $\neq$ NP, some reductions must be inherently expensive, forcing Ω to grow without bound and creating unbounded curvature.

Theorem 16 (Geometric Separation Criterion)**.** *If $P \neq NP$, then $\mathcal{M}_K$ exhibits at least one of: (i) unbounded curvature, (ii) infinite diameter, (iii) non-trivial π_1. If $P = NP$, then $\mathcal{M}_K$ is a bounded flat space.*

This is one of the most far-reaching results of the paper. It provides a geometric "litmus test" for P vs NP. If you could prove that the KARMA manifold has unbounded curvature (or infinite diameter, or non-trivial fundamental group), that would imply P $\neq$ NP. Conversely, if P = NP, then the manifold must be a tame, bounded, flat object. The theorem does not resolve P vs NP, but it provides a new angle of attack: proving geometric properties of a manifold rather than proving time bounds for algorithms.

Proof. P = NP implies all reductions cost $O(n^k)$ for universal k, yielding bounded Ω, curvature, and diameter. P $\neq$ NP (under ETH [12]) implies superpolynomial growth of T_{red} for some instances, forcing at least one of (i)–(iii). □

The forward direction (P = NP $\Rightarrow$ flat) follows from the observation that if all problems are solvable in polynomial time, then all reductions between them also run in polynomial time, making Ω bounded and the curvature bounded. The converse direction (P $\neq$ NP $\Rightarrow$ geometric complexity) relies on the Exponential Time Hypothesis (ETH), which states that some NP-complete problems require exponential time. Under ETH, the reduction time T_{red} must grow superpolynomially for some instances, forcing Ω to grow without bound and creating unbounded curvature or infinite diameter.

Conjecture 17 (Geometric P $\neq$ NP)**.** $\sup_{p \in \mathcal{M}_K} |R(p)| = +\infty$*, which would imply $P \neq NP$ under Theorem 16.*

We conjecture that the KARMA manifold has unbounded curvature—that is, for any bound B, there exist problem instances where the scalar curvature exceeds B. If proven, this would establish P $\neq$ NP as a corollary of Theorem 16. The conjecture is motivated by the observation that reduction complexity grows polynomially with instance size, and the curvature formula involves derivatives of T_{red}, which should grow without bound as $n \to \infty$.

 doi:10.20944/preprints202603.0871.v1

9.2. Curvature Complexity Classes

Traditional complexity theory classifies problems into coarse categories: P, NP, PSPACE, etc. The KARMA framework enables a finer-grained classification based on curvature. We define NP$[\kappa]$ as the class of NP problems whose associated manifold has curvature bounded by κ. This creates an infinite hierarchy of classes between P and NP, providing a continuous "hardness spectrum" rather than a binary easy/hard distinction.

Definition 10 (Curvature Classes). $NP[\kappa] = \{\Pi \in NP : \sup_p |R(p)| \leq \kappa\}$.

NP$[\kappa]$ is the set of NP problems whose computational landscape has curvature at most κ. Problems in NP$[0]$ live on a flat manifold (uniformly efficient reductions), while problems with high curvature bounds are in NP$[\kappa]$ for large κ. This creates a fine-grained "ruler" for measuring how hard a problem is, going beyond the binary P/NP distinction.

Proposition 4 (Hierarchy). $P \subseteq NP[0] \subsetneq NP[\kappa_1] \subsetneq NP[\kappa_2] \subsetneq \cdots \subsetneq NP$ *for* $0 < \kappa_1 < \kappa_2 < \cdots$.

There is a strict hierarchy of curvature classes, with P at the bottom (flattest) and full NP at the top (most curved). Each step up the hierarchy includes problems with higher curvature, i.e., problems where reduction costs vary more sharply. This provides a geometric analogue of the polynomial hierarchy in traditional complexity theory.

9.3. Information–Complexity Duality

We establish a fundamental connection between information theory and computational complexity. The key insight is that reductions that "compress" information (mapping many different inputs to the same output) must take proportionally more time. This is a geometric version of the pigeonhole principle: if a reduction is highly non-injective (its fibers are large), then distinguishing which original instance was mapped requires significant computation.

Proposition 5 (Duality). $T_{ij}(n) = \Omega(S_{ij})$, *where* $S_{ij} = \log \mathrm{Vol}(\tau_{ij}^{-1}(q))$ *measures information loss.*

The time required for a reduction from Π_i to Π_j is at least proportional to the "entropy" S_{ij}—the logarithm of the number of Π_i-instances that map to the same Π_j-instance. Reductions that compress more information (larger S_{ij}) must take more time. You cannot compress information for free—this is a fundamental constraint that connects the geometry of the manifold to the thermodynamics of computation.

Reductions that compress information (many inputs $\to$ one output) must take proportionally more time. You cannot compress information for free.

 doi:10.20944/preprints202603.0871.v1

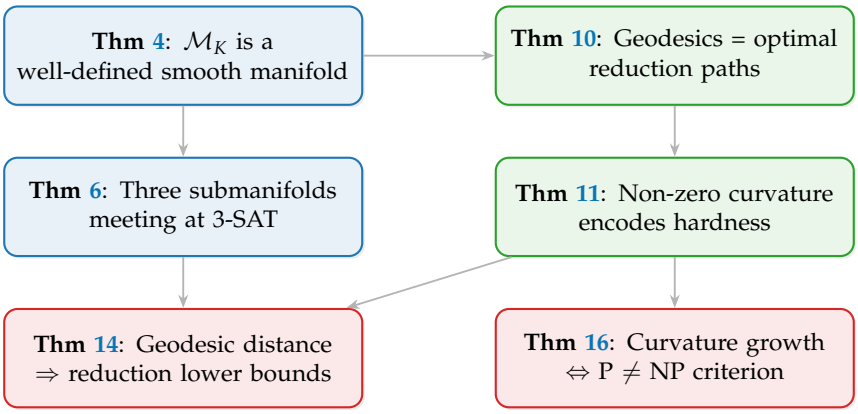

Figure 14. Logical dependencies among key results. The manifold construction (Thm 4) enables both the categorical decomposition (Thm 6) and the Riemannian structure (Thm 10). These in turn yield the curvature-hardness correspondence (Thm 11), reduction lower bounds (Thm 14), and the geometric P vs NP criterion (Thm 16).

10. Limitations and Future Work

Every theoretical framework has limitations, and we believe it is important to state them clearly. This section identifies the main assumptions and approximations in the KARMA construction, explains why they are necessary in the current formulation, and suggests how future work might overcome them. We also outline promising research directions that the framework opens up.

Mollification approximation. Smooth transitions use mollification with parameter ϵ; the $\epsilon \to 0$ limit requires further analysis. *In plain terms:* we smoothed out the "sharp corners" of discrete reductions to make them infinitely differentiable. While this preserves correctness on actual problem instances, the behaviour between instances is an approximation. Understanding the limiting behaviour as $\epsilon \to 0$ is an important open question.

Finite sample. Only 21 of ∼3000 known NP-complete problems are modelled; the full manifold may be considerably richer. *In plain terms:* Karp's 21 problems are a representative sample, but thousands more NP-complete problems have been discovered since 1972. Extending the manifold to include all known NP-complete problems might reveal additional categorical submanifolds, richer topology, and more complex curvature patterns.

Instance-size dependence. The metric depends on n, creating a family $\{\mathcal{M}_K^{(n)}\}$ rather than a universal manifold. *In plain terms:* the manifold we construct depends on the size of the problem instances being considered. A graph on 10 vertices lives on a different manifold than a graph on 1000 vertices. Ideally, one would like a single "universal" manifold encompassing all sizes simultaneously; constructing such an object (e.g., as an inverse limit) is an open mathematical challenge.

Free parameter. The conformal factor involves $\lambda > 0$; qualitative results are robust but optimal calibration is open. *In plain terms:* the parameter λ controls how much weight computational cost receives in the metric. All our qualitative results (non-zero curvature, positive sectional curvature, etc.) hold for any $\lambda > 0$, but quantitative predictions (exact distances, precise curvature values) depend on the choice of λ. Calibrating λ against empirical reduction benchmarks is future work.

Future directions. Extension to all NP-complete problems; quantum KARMA for QMA-complete problems; gradient flows as adaptive solver models; parameterized complexity via fibrations; and a software library for manifold-aware solver design.

11. Conclusion

This paper introduced the **Karp Algebraic Reduction Manifold Architecture (KARMA)**, a geometric framework that brings together Karp's 21 classical NP-complete problems within a single unified structure. Traditionally, these problems are studied as separate puzzles connected by chains

 doi:10.20944/preprints202603.0871.v1

of polynomial-time reductions. In contrast, KARMA shows that they can be interpreted as different coordinate views of the same underlying mathematical landscape of computational difficulty.

In this framework, each NP-complete problem corresponds to a coordinate chart on a smooth manifold, while reductions between problems appear as smooth transitions between charts. The problems naturally organize into three broad families—graph problems, set problems, and number problems—which form structured regions of this space and intersect at a central junction represented by the well-known 3-SAT problem.

This geometric viewpoint provides new interpretations of computational complexity. Distances on the manifold represent the effort required to transform one problem into another, and the shortest paths correspond to the most efficient reduction sequences. The curvature of the space reflects variations in computational hardness across the landscape.

An important consequence is the **single-reduction propagation principle**: a reduction from any new problem to one NP-complete problem automatically extends to all others through the manifold's transition functions. By revealing this hidden structure, KARMA offers a new conceptual lens for understanding computational complexity and may guide future research in algorithms, complexity theory, and automated problem transformations.

The unification of Karp's 21 problems into a single geometric structure suggests that beneath the surface complexity of individual problems lies a fundamental mathematical simplicity—the simplicity of differential geometry. Whether this geometric perspective will unlock the P versus NP question remains to be seen, but it enriches our understanding of what it means for problems to be "computationally equivalent."

Acknowledgments: This publication has emanated from research supported by a grant from Research Ireland under Grant number 12-RC-2289-P2, which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

References

1. Cook, S.A. The complexity of theorem-proving procedures. *Proceedings of the third annual ACM symposium on Theory of computing* **1971**, pp. 151–158.
2. Karp, R.M. Reducibility among combinatorial problems. In Proceedings of the Complexity of computer computations. Springer, 1972, pp. 85–103.
3. Garey, M.R.; Johnson, D.S. *Computers and intractability: A guide to the theory of NP-completeness*; W. H. Freeman and Company, 1979.
4. Johnson, D.S. Computers and intractability: A guide to the theory of NP-completeness. *WH Freeman and Company, San Francisco* **1979**.
5. Mulmuley, K.D.; Sohoni, M. Geometric complexity theory I: An approach to the P vs. NP and related problems. *SIAM Journal on Computing* **2001**, *31*, 496–526.
6. Abramsky, S.; Coecke, B. A categorical semantics of quantum protocols. *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science* **2004**, pp. 415–425.
7. Lee, J.M. *Introduction to smooth manifolds*, 2nd ed.; Springer Science & Business Media, 2012.
8. Amari, S.i. *Information geometry and its applications*; Springer, 2016.
9. Do Carmo, M.P. *Riemannian geometry*; Birkhäuser, 1992.
10. Jost, J. *Riemannian geometry and geometric analysis*, 7th ed.; Springer, 2017.
11. Petersen, P. *Riemannian geometry*, 2nd ed.; Springer, 2006.
12. Impagliazzo, R.; Paturi, R. On the complexity of k-SAT. *Journal of Computer and System Sciences* **2001**, *62*, 367–375.