Abstract

Current artificial intelligence benchmarks primarily evaluate a model's ability to recognize or predict observable states. Across domains such as medicine, biology, engineering, and risk prediction, performance is typically assessed through a model's ability to classify, predict, rank, or forecast outcomes. These tasks measure whether a model can recognize patterns associated with a target state, but they provide limited insight into whether a model can identify the underlying structures that generate those states.

Many real-world systems are governed by interacting constraints, compensatory processes, delayed effects, incomplete observations, competing mechanisms, and shifting sources of dominance. In such settings, similar outcomes may arise from different governing structures, while similar structures may produce different outcomes depending on context and system dynamics. When intervention decisions depend on understanding why a system behaves as it does, outcome prediction alone may be insufficient.

We introduce Constraint Geometry Benchmarks, a benchmark category designed to evaluate whether AI systems can distinguish among alternative governing structures that produce overlapping or identical outcomes. The framework focuses on tasks involving interacting constraints, compensation, partial observability, signal degradation, dominance relationships, intervention structure, and mechanism attribution. Rather than asking only whether a model can predict an outcome, the benchmark evaluates whether it can identify the structural configuration most consistent with the available evidence.

For example, a benchmark instance may present partially observed trajectories that converge on the same outcome despite arising from different constraint structures. The task is not to predict the outcome, which is already known, but to determine which governing mechanism best explains the observed trajectory. Such tasks evaluate structural attribution rather than state recognition.

This distinguishes Constraint Geometry Benchmarks from outcome-prediction benchmarks, mechanistic scientific models, and causal discovery benchmarks alike by focusing on constraint discrimination under partial observability rather than forecasting, biological etiology, or full generative-structure recovery.

To demonstrate the framework, we introduce the Constraint Geometry Benchmark Suite (CGBS), a family of synthetic oncology-inspired benchmark datasets. Oncology serves as a useful demonstration environment because it naturally contains interacting repair, immune, inflammatory, metabolic, and signaling processes operating under incomplete observability and dynamic constraint competition. The benchmark suite is designed to reduce shortcut learning, single-variable separation, direct label leakage, and other common benchmark artifacts. Models are evaluated not only on outcome prediction but also on Structural Alignment Accuracy, a measure of whether the governing mechanism associated with an observed trajectory is correctly identified.

The primary contribution of this work is a benchmark-generation methodology for evaluating mechanism attribution under controlled conditions. We do not propose a biological theory, clinical model, diagnostic framework, or treatment system, nor do we claim that success on synthetic structural reasoning tasks implies successful reasoning in real-world domains. We further distinguish between structural reasoning under known generative rules, which is the focus of this framework, and structural discovery under unknown rules, which remains an open research problem.

More broadly, we argue that benchmark evaluation can be extended beyond state recognition to assess whether models can discriminate among competing generative structures. Constraint Geometry is offered as a framework for investigating whether mechanism attribution constitutes a distinct and measurable capability in artificial intelligence systems. Rather than assuming that mechanism attribution constitutes a distinct capability, the benchmark suite provides a controlled environment in which to test whether it is separable from conventional pattern recognition.

Executive Summary

The Benchmark Gap

Most AI benchmarks evaluate a model's ability to recognize, classify, predict, rank, or forecast observable states.

Across domains such as medicine, biology, engineering, finance, and general reasoning, performance is typically measured by comparing model outputs against known labels or future outcomes. These benchmarks have proven highly effective at assessing pattern recognition and predictive accuracy.

What they reveal less clearly is whether a model understands the structures that generate the outcomes it predicts.

In many real-world systems, similar outcomes can arise from different underlying mechanisms, while similar mechanisms can produce different outcomes depending on context, compensation, delays, hidden variables, and competing influences. As a result, accurate prediction does not necessarily imply accurate understanding.

This paper explores a simple question:

Is recognizing a state the same as identifying the structure that produced it?

We introduce a benchmark framework designed to evaluate that distinction directly.

Constraint Geometry in Brief

Constraint Geometry is a benchmark framework for evaluating mechanism attribution under controlled conditions.

The framework starts from the observation that many complex systems are shaped not by isolated variables, but by interactions among constraints. System behavior emerges through competing demands, compensatory processes, delayed effects, incomplete information, and shifting sources of dominance.

Rather than asking whether a model can predict an outcome, Constraint Geometry asks whether it can identify the structural configuration most consistent with the available evidence.

The goal is to determine whether models can distinguish among competing explanations that produce similar observable behavior.

Constraint Geometry is a benchmark methodology, not a theory of cancer, a clinical model, a diagnostic framework, or a causal discovery system.

What the Benchmark Measures

Constraint Geometry focuses on recurring dimensions of structural reasoning:

Constraint Interaction

Can a model identify how multiple constraints jointly shape system behavior?

Compensation

Can a model recognize when one process temporarily masks instability elsewhere?

Partial Observability

Can a model reason effectively when important information is unavailable?

Signal Degradation

Can a model distinguish between missing information and deteriorating information?

Dominance

Can a model determine which constraint currently exerts the greatest influence on system trajectory?

Intervention Structure

Can a model identify which intervention is most aligned with the governing constraint?

Mechanism Attribution

Can a model distinguish among competing explanations that produce similar outcomes?

Together, these dimensions define a benchmark category aimed at evaluating structural attribution rather than state recognition alone.

Benchmark Architecture

To investigate these questions, we introduce the Constraint Geometry Benchmark Suite (CGBS), a family of synthetic oncology-inspired benchmarks designed to evaluate structural reasoning under controlled conditions.

The suite is organized as a progression of increasingly demanding structural challenges.

v0.1 Constraint Geometry

Primary Question: Is instability present within the system?

Focus: Interacting constraints and stability classification.

v0.2 Compensation Geometry

Primary Question: Is compensation masking instability?

Focus: Detection of hidden instability despite apparent stability.

v0.3 Readability Collapse

Primary Question: Can the system still correctly interpret critical signals?

Focus: Signal interpretation failure and information degradation.

v0.4 Signal Alignment

Primary Question: Are available signals aligned with system requirements?

Focus: Alignment between observed information and structural need.

v0.5 Signal Latency Boundary

Primary Question: Has information delay exceeded recoverable limits?

Focus: Timing-dependent failure and delayed intervention effects.

v0.6 Missing Signal Detection

Primary Question: Is a critical signal absent?

Focus: Reasoning under incomplete information.

v0.7 Constraint Dominance

Primary Question: Which constraint governs the trajectory of the system?

Focus: Attribution of behavior to dominant governing mechanisms.

The architecture progressively increases structural complexity while reducing opportunities for shortcut learning, label leakage, proxy prediction, and single-variable separation.

Evaluation occurs on two levels:

Target A: Outcome prediction

Target B: Mechanism attribution

Models are assessed using conventional predictive metrics alongside **Structural Alignment Accuracy (SAA)**, which measures whether the governing structural explanation associated with an observed trajectory is correctly identified.

Core Propositions

This work advances five propositions.

Proposition 1

State recognition and mechanism attribution may represent distinct evaluation targets.

Proposition 2

Outcome prediction alone may be insufficient for assessing intervention-relevant reasoning in complex systems.

Proposition 3

Synthetic benchmark environments can be constructed to separate mechanism attribution from straightforward outcome prediction.

Proposition 4

Benchmark evaluation can be extended beyond state recognition toward discrimination among competing structural explanations.

Proposition 5

Whether mechanism attribution constitutes a distinct capability in AI systems remains an empirical question rather than an established conclusion.

The purpose of CGBS is to provide a controlled environment in which that question can be tested.

Implications

If successful, this framework suggests a broader class of benchmark tasks that extend beyond outcome prediction.

Biomedical AI

Benchmarks could evaluate whether systems distinguish among competing disease mechanisms rather than merely predict clinical outcomes.

Safety-Critical Systems

Benchmarks could evaluate whether systems identify governing failure modes before intervention decisions are made.

General AI Evaluation

Benchmarks could investigate whether structural attribution and mechanism discrimination are measurable capabilities distinct from pattern recognition and state classification.

These examples are illustrative rather than exhaustive. The contribution is a benchmark design framework, not a comprehensive theory of reasoning evaluation.

Scope and Limits

The contribution of this work is a benchmark-generation methodology.

It is not a biological theory, clinical framework, diagnostic system, or treatment model. It does not establish the true mechanisms of disease, nor does it demonstrate transfer from synthetic benchmark performance to real-world reasoning performance.

The framework evaluates mechanism attribution under known generative rules within controlled benchmark environments. We distinguish this setting from structural discovery under unknown rules, which remains an open research problem.

At a high level, traditional benchmarks primarily ask:

"What state is the system in?"

Constraint Geometry asks:

"What structure governs the trajectory of the system?"

Whether that distinction ultimately corresponds to a distinct capability in artificial intelligence remains an empirical question. The benchmark framework introduced here is intended to provide a controlled environment in which that question can be rigorously evaluated.

Part I — The Benchmark Gap

1. Introduction

Over the past decade, benchmark-driven evaluation has become one of the dominant methodologies for measuring progress in artificial intelligence. Improvements in benchmark performance are frequently used as evidence of advances in reasoning, prediction, understanding, and general capability. Across scientific, medical, engineering, and operational domains, benchmark scores increasingly shape how systems are compared, validated, and deployed.

This approach has produced substantial benefits. Standardized benchmarks have accelerated model development, enabled reproducible comparisons, and created shared evaluation targets across research communities. Tasks that once required extensive human interpretation can now be assessed using large-scale benchmark suites and quantitative performance metrics.

At the same time, benchmark design inevitably influences which capabilities are measured and which remain invisible. Every benchmark embodies assumptions about what constitutes successful performance. A benchmark focused on classification rewards classification ability. A benchmark focused on prediction rewards predictive accuracy. A benchmark focused on retrieval rewards information recovery. As benchmark performance improves, an important question emerges: what capabilities are being measured, and what capabilities remain largely unevaluated?

This paper focuses on one possible gap within current benchmark ecosystems. Specifically, it examines whether existing benchmarks adequately evaluate a model's ability to distinguish among competing structural explanations for observed outcomes. While many benchmarks measure whether a model can recognize a state, relatively few are explicitly designed to evaluate whether a model can identify the governing structure responsible for that state.

The distinction is subtle but potentially important. A system may correctly predict an outcome while misidentifying the mechanism that produces it. Likewise, multiple mechanisms may generate similar observable outcomes, making successful prediction insufficient for determining whether a model has correctly understood the underlying structure of a problem.

The sections that follow examine how contemporary benchmarks have evolved around state recognition tasks, what forms of reasoning they typically measure, and where a potential category of structural evaluation may remain underdeveloped.

1.1 The Rise of State Classification

Many of the most influential benchmark tasks in contemporary artificial intelligence are organized around identifying, predicting, or classifying states.

In medicine, models are evaluated on diagnostic classification, prognosis estimation, treatment-response prediction, disease progression forecasting, and risk assessment. Similar patterns appear in engineering, finance, climate science, and operational decision systems, where benchmark success is frequently defined by accurate prediction of observable outcomes.

Representative examples include:

- Diagnosis
- Prognosis
- Survival prediction
- Treatment-response prediction
- Risk estimation
- Failure prediction
- Event forecasting
- State classification

Although these tasks differ in content and methodology, they share a common structure. A model receives observations and is evaluated according to its ability to identify or predict a target state.

This paradigm has been remarkably successful. State-oriented benchmarks have driven substantial improvements in predictive performance across many domains. In numerous applications, accurate state recognition provides significant practical value and remains an essential component of intelligent behavior.

The success of these benchmarks, however, does not necessarily imply that all relevant forms of reasoning are being evaluated. Benchmark performance may reveal that a model can recognize patterns associated with an outcome without revealing whether it understands the structural relationships that produced that outcome.

This distinction becomes increasingly relevant as AI systems are asked not only to predict outcomes but also to support explanations, interventions, planning, and decision-making in complex environments.

1.2 What Existing Benchmarks Measure

Most benchmark families evaluate one or more forms of state-oriented reasoning.

Common benchmark targets include:

- Recognition
- Classification
- Prediction
- Ranking
- Correlation extraction
- Pattern matching
- Outcome forecasting

These capabilities are valuable and often necessary. Successful performance frequently requires substantial information processing, abstraction, and generalization.

Importantly, this paper does not argue that such benchmarks are inadequate or uninformative. On the contrary, state-recognition benchmarks have played a central role in advancing artificial intelligence.

The question is not whether these benchmarks measure meaningful capabilities.

The question is whether they measure all capabilities relevant to understanding complex systems.

Many benchmark tasks can be solved by learning associations between observations and outcomes. A model may achieve high performance by identifying statistical regularities,

predictive features, or latent patterns that correlate with the correct answer.

In such settings, benchmark success demonstrates the ability to predict or classify outcomes.

What remains less clear is whether success also demonstrates understanding of the structural relationships responsible for generating those outcomes.

This distinction becomes especially important when multiple underlying mechanisms can produce similar observable states.

1.3 What Existing Benchmarks Rarely Measure

While benchmark ecosystems have become increasingly sophisticated, several forms of structural evaluation remain comparatively uncommon.

These include:

Governing Constraints

Which factors exert the strongest influence on system behavior?

Compensation

Which processes are masking instability or failure elsewhere in the system?

Dominance

Which constraint currently governs the trajectory of the system?

Mechanism Attribution

Which explanation best accounts for the observed outcome?

Structural Failure Geometry

How do interacting constraints combine to produce instability, degradation, or collapse?

These questions differ from traditional prediction tasks.

They require distinguishing among competing structural explanations rather than identifying a single observable outcome.

Two systems may produce identical outcomes while being governed by different constraints.

Two failures may appear identical while arising from different mechanisms.

Two trajectories may converge on the same endpoint while requiring different interventions.

In each case, outcome recognition alone may be insufficient to determine which structural explanation is correct.

This observation does not imply that existing benchmarks are flawed. Rather, it suggests the possibility that a distinct category of benchmark task may exist—one focused on evaluating whether models can reason about governing structures rather than simply recognize resulting states.

1.4 The Missing Benchmark Category

The motivation for this work can be summarized by a distinction between two benchmark questions:

What state is present?

and

What structure produced that state?

In many settings, these questions are closely related. A correctly identified state often provides useful information about the processes that generated it. The relationship, however, is not one-to-one.

Different structures can produce similar outcomes. Different mechanisms can generate similar observations. Compensatory processes can conceal instability. Partial observability can obscure critical causes. Dominance relationships can shift over time. Consequently, successful state recognition does not necessarily imply successful mechanism attribution.

This observation points toward a benchmark category that is largely distinct from both traditional prediction tasks and full causal discovery. Traditional benchmarks typically evaluate whether a model can recognize, classify, or predict an outcome. Causal discovery methods aim to recover unknown generative structure from data. Between these extremes lies a different class of problems: determining which of several competing structural explanations is most consistent with the available evidence.

This middle space includes tasks in which multiple candidate mechanisms remain plausible, observations are incomplete, constraints interact, and direct recovery of the underlying generative process is neither required nor possible. The objective is not to infer an entire causal graph, but to discriminate among competing structural explanations under conditions of ambiguity.

Benchmarks in causal discovery, counterfactual reasoning, and mechanistic analysis address related questions, but they generally focus on structure recovery, intervention effects, or known generative relationships. The benchmark category proposed here focuses instead on mechanism attribution under partial observability and competing constraints.

The central claim of this paper is not that mechanism attribution has already been established as a distinct capability. Rather, it is that the distinction between state recognition and mechanism attribution may itself be measurable through carefully constructed benchmark tasks. If so, benchmark evaluation can be extended beyond the question of whether a model predicts the correct outcome to include whether it identifies the correct structural explanation for that outcome.

The remainder of this paper introduces a benchmark framework designed to investigate that possibility.

2. From State Recognition to Mechanism Attribution

Contemporary benchmark ecosystems are primarily organized around state recognition. Across domains, models are evaluated according to their ability to classify, predict, rank, retrieve, or forecast observable outcomes. These tasks have driven substantial advances in artificial intelligence and remain essential components of intelligent behavior.

This paper explores a complementary evaluation target: **mechanism attribution**.

Rather than asking whether a model can identify an outcome, mechanism attribution asks whether it can identify the structural explanation most consistent with the available evidence. The distinction matters because similar outcomes can arise from different underlying mechanisms, while similar mechanisms can produce different outcomes depending on context, compensation, timing, and observation constraints.

As a result, predictive success alone may be insufficient to determine whether a model has correctly identified the structure responsible for an observed trajectory.

The central question motivating this work is therefore straightforward:

Is recognizing a state the same as identifying the structure that produced it?

In some settings, the answer may be yes. In others, multiple structural explanations may remain consistent with the same observations. If so, benchmark evaluation may benefit from distinguishing between state recognition and mechanism attribution as separate evaluation targets.

This section develops that distinction, situates it within existing systems-oriented perspectives, clarifies the scope of the proposed framework, and defines the benchmark question that motivates the remainder of the paper.

2.1 Systems-Oriented Perspectives

The idea that system behavior emerges from interactions among components rather than isolated variables is well established across multiple scientific traditions.

Systems biology emphasizes networks of interacting processes operating across multiple scales rather than isolated genes, proteins, or pathways. Network medicine similarly views disease as a property of interconnected biological systems rather than individual components. Robustness theory highlights the role of compensation, redundancy, and adaptation in maintaining stability despite perturbation. Research on attractor dynamics examines how different trajectories may converge on similar outcomes, while complex adaptive systems theory emphasizes emergence, feedback, nonlinearity, and distributed interactions.

Despite their differences, these traditions share a common observation: observable behavior is often shaped by relationships among interacting processes rather than by isolated variables.

This observation has important implications for evaluation. If system behavior emerges from interactions among constraints, feedback loops, compensatory processes, and competing influences, then outcome recognition alone may not fully capture whether a model has identified the structures responsible for generating those outcomes.

Constraint Geometry does not seek to extend or replace these traditions. Instead, it draws upon a common insight that underlies them and asks a benchmark-oriented question:

Can benchmark evaluation assess whether a model correctly attributes observed behavior to its governing structure?

2.2 Why Mechanism Attribution Matters

Many practical decisions depend not only on what outcome is occurring, but on why it is occurring.

Different mechanisms may produce similar observable states while requiring different interventions. A system that appears stable may be sustained by compensatory processes that conceal underlying instability. Two trajectories may converge on the same outcome despite being governed by different constraints. Likewise, identical interventions may produce different effects when applied to systems governed by different structural conditions.

In such settings, accurate prediction and accurate attribution are related but not identical capabilities.

A model may correctly predict an outcome while misidentifying the mechanism responsible for it. Conversely, a model may correctly identify the governing mechanism even when outcome prediction remains uncertain due to incomplete information or stochastic dynamics.

This observation does not imply that mechanism attribution is always necessary, nor does it diminish the importance of prediction. Rather, it suggests that outcome prediction and mechanism attribution may represent distinct dimensions of capability that can be evaluated separately.

The purpose of Constraint Geometry is to investigate whether that distinction can be measured empirically.

2.3 What Constraint Geometry Is Not

Because Constraint Geometry uses oncology-inspired benchmark environments and draws conceptual inspiration from systems-oriented disciplines, its scope requires careful definition.

Constraint Geometry is not a theory of cancer, a biological model, a clinical framework, or a treatment methodology. The benchmark environments used throughout this work are synthetic and are not intended to establish biological truth.

Nor is Constraint Geometry a replacement for systems biology, network medicine, robustness theory, or other systems-oriented approaches. Those fields seek to explain the organization and behavior of real-world systems. Constraint Geometry seeks to evaluate AI systems operating within controlled benchmark environments.

Most importantly, Constraint Geometry is not a causal discovery framework. It does not attempt to recover complete causal structure from observational data or infer unknown generative graphs from scratch.

Instead, it focuses on a narrower and more tractable problem: determining which of several competing structural explanations is most consistent with the available evidence.

This distinction is central to the benchmark framework.

The objective is not structural discovery under open-ended uncertainty. The objective is mechanism attribution under controlled conditions.

2.4 What Constraint Geometry Is

Constraint Geometry defines a benchmark category centered on mechanism attribution.

In conventional benchmark tasks, models are evaluated according to whether they identify the correct outcome. In Constraint Geometry tasks, the outcome may already be known. The primary challenge is to determine which structural explanation best accounts for the observed trajectory.

The benchmark therefore shifts evaluation from outcome recognition toward discrimination among competing generative explanations.

This middle ground occupies a space between traditional prediction tasks and full causal discovery. Traditional prediction asks whether a model can recognize or forecast an outcome. Causal discovery seeks to recover unknown generative structure. Constraint Geometry instead evaluates whether a model can distinguish among plausible structural explanations that remain consistent with the available evidence.

The framework is designed for settings characterized by partial observability, interacting constraints, compensation, signal degradation, delayed effects, and shifting dominance

relationships. Within such environments, multiple mechanisms may remain plausible even when the outcome itself is known.

Constraint Geometry evaluates whether a model can identify which explanation is most structurally aligned with the observed evidence.

In this sense, Constraint Geometry is best understood as an evaluation methodology rather than a scientific theory. Its purpose is to create controlled benchmark environments in which mechanism attribution can be measured, compared, and analyzed.

2.5 Core Hypothesis

The central hypothesis of this paper concerns benchmark evaluation rather than system behavior.

Contemporary benchmarks primarily evaluate state recognition. Examples include:

- What diagnosis is present?
- What outcome is most likely?
- What risk category applies?
- What event will occur next?

These are important and valuable benchmark targets.

Constraint Geometry introduces a complementary target:

What structure governs the trajectory of the system?

State recognition concerns observable outcomes.

Mechanism attribution concerns structural explanations.

The two capabilities may overlap, but they are not necessarily identical.

A model may correctly identify an outcome while failing to identify the mechanism responsible for it. Conversely, a model may correctly identify a governing mechanism even when outcome prediction remains uncertain.

The hypothesis explored throughout this paper is therefore not that mechanism attribution is already known to be a distinct capability. Rather, it is that the distinction between state recognition and mechanism attribution may itself be measurable through appropriately designed benchmark tasks.

2.6 The Benchmark Question

The difference between traditional benchmark evaluation and Constraint Geometry can be summarized through a shift in the questions being asked.

Traditional Benchmark Question	Constraint Geometry Question
What state is observed?	What structure produced that state?
What outcome is most likely?	Which mechanism best explains the outcome?
Will the system fail?	What constraint governs failure?
Will recovery occur?	What prevents recovery?
Which label is correct?	Which mechanism generated the label?

Constraint Geometry does not replace prediction, classification, forecasting, or state recognition. Those capabilities remain essential and continue to represent important benchmark targets.

The proposal advanced in this paper is that benchmark ecosystems may benefit from an additional layer of evaluation: assessing whether models can correctly attribute observed behavior to the structural mechanisms that generate it.

The remainder of this paper introduces a benchmark architecture designed to investigate that possibility.

3. From Variables to Relationships

Most machine learning systems are trained to map observations to outcomes.

Inputs are represented as features, variables, measurements, or tokens. Models learn statistical relationships between those inputs and target labels. Success is measured by how accurately outcomes can be classified, predicted, ranked, or forecast.

This paradigm has been extraordinarily successful. Modern machine learning systems routinely identify patterns that would be difficult or impossible to discover through manual analysis alone.

Yet many real-world systems are not governed by variables in isolation. Their behavior emerges from interactions among multiple processes operating simultaneously across time. Stability, failure, adaptation, and recovery often depend less on the value of any single variable than on the relationships among variables.

This distinction motivates the conceptual foundation of Constraint Geometry.

The framework does not reject variables. Variables remain essential components of any analytical system. Rather, it proposes that an important class of reasoning problems concerns how variables interact, compete, compensate, align, and constrain one another.

Under this view, the primary object of analysis is not the variable itself, but the structure formed by relationships among variables.

Constraint Geometry is built around the hypothesis that benchmark tasks can be designed to evaluate whether models identify those structures when performing mechanism attribution.

3.1 Variable-Centric Reasoning

Most contemporary machine learning systems are fundamentally variable-centric.

A model receives observations represented as variables and learns associations between those observations and desired outcomes. In medical applications, these variables may include laboratory values, biomarkers, imaging findings, symptoms, demographic characteristics, or physiological measurements. Similar structures appear across engineering, finance, climate science, and operational decision systems.

The objective is straightforward: identify patterns that improve prediction.

Examples include:

- Disease present or absent
- Survival or mortality
- Response or non-response
- High risk or low risk

The resulting models may achieve impressive predictive performance.

Importantly, however, successful prediction does not require explicit representation of the mechanism responsible for the outcome. A model need only identify patterns that reliably correlate with the correct answer.

This is not a limitation of machine learning. Correlation-based prediction often provides substantial practical value.

The challenge arises when different mechanisms produce similar observational signatures. In such settings, prediction accuracy alone may provide limited evidence that a model has identified the structural explanation responsible for the outcome.

Variable-centric reasoning therefore emphasizes relationships between observations and outcomes.

Constraint Geometry asks whether benchmark evaluation can assess an additional layer of reasoning: relationships among the observations themselves.

3.2 Relationship-Centric Reasoning

Relationship-centric reasoning shifts the focus of analysis.

The question is no longer:

Which variables are present?

Instead, it becomes:

How do the variables relate to one another?

Within complex systems, individual measurements often derive meaning from their relationships to other measurements.

Consider a simplified example.

A repair process may appear normal when viewed independently. An inflammatory process may also appear normal when viewed independently. Yet the relationship between repair demand and repair capacity may reveal a growing mismatch that neither variable captures alone.

The individual measurements remain within expected ranges.

The relationship does not.

In such cases, the structure governing system behavior resides not in the variables themselves but in the interactions among them.

Relationship-centric reasoning therefore emphasizes concepts such as:

- Coupling
- Alignment
- Competition
- Compensation
- Dominance
- Feedback
- Dependency

These concepts describe patterns of interaction rather than isolated measurements.

Constraint Geometry is founded on the premise that benchmark tasks can be constructed to evaluate whether models identify such interaction patterns when attributing mechanisms.

The goal is not to replace variable-based analysis. The goal is to determine whether benchmark evaluation can measure a complementary capability centered on structural relationships.

3.3 Relationships as Constraints

Constraint Geometry treats relationships as constraints on system behavior.

A constraint is not merely a variable. It is a condition that shapes the range of trajectories available to a system.

Some constraints enable behavior. Others limit it. Many interact.

A repair process may constrain the effects of damage. Resource availability may constrain adaptation. Information flow may constrain coordination. Timing may constrain recovery.

System behavior emerges from the interaction of these constraints rather than from any single constraint considered independently.

This perspective shifts attention from isolated measurements to patterns of influence.

The benchmark question therefore becomes:

Which constraints govern the observed trajectory?

Rather than:

Which variable has changed?

This distinction is central to the framework.

Mechanism attribution requires identifying the constraint structure most consistent with the available evidence rather than merely identifying the variables associated with an outcome.

3.4 Stability as Geometry

The term *Constraint Geometry* reflects a particular view of stability.

Traditional prediction tasks focus primarily on outcomes.

Constraint Geometry focuses on trajectories.

A trajectory describes how a system evolves as interacting constraints shape its future states.

Under this perspective, stability is not treated as a static condition. It is treated as an emergent property of relationships among constraints.

A system remains stable when interacting constraints remain sufficiently aligned to sustain function.

A system becomes unstable when those relationships degrade beyond recoverable limits.

Importantly, instability may emerge without any individual variable crossing an obvious threshold.

Instead, instability may arise through:

- Growing misalignment
- Accumulating delay
- Compensation failure
- Constraint competition
- Signal degradation
- Dominance shifts

These processes alter the structure of the system before they necessarily alter any single measurement.

The geometry metaphor captures this idea.

Just as the shape of an object depends on relationships among points rather than any individual point, system behavior may depend on relationships among constraints rather than any individual variable.

Constraint Geometry therefore treats stability, instability, recovery, and collapse as properties of relational structure.

3.5 Why Relationships Matter

The practical motivation for relationship-centric evaluation is simple.

Different mechanisms can produce similar outcomes.

Similar outcomes can require different interventions.

Consider two systems that exhibit the same observable failure.

In one system, failure emerges because compensatory processes have been exhausted. In another, failure emerges because critical signals no longer reach the processes responsible for adaptation.

The outcome is similar.

The governing mechanisms are not.

A model focused exclusively on outcome recognition may treat these situations as equivalent. A model capable of mechanism attribution would distinguish between them.

The distinction matters because interventions operate on mechanisms rather than outcomes.

Restoring compensation capacity and restoring signal transmission are different actions, even when the observed failure appears identical.

The same principle applies to recovery. Two systems may appear equally unstable while differing substantially in their underlying structural conditions. One may remain recoverable because compensatory capacity is intact. The other may be approaching irreversible collapse because that capacity has already been exhausted.

The observable state alone may not reveal the difference.

The structural relationships may.

This observation motivates the benchmark framework developed throughout the remainder of this paper.

If different mechanisms can generate similar outcomes, and if effective intervention depends on distinguishing among those mechanisms, then benchmark evaluation may benefit from

assessing whether models can identify the structural explanations that produce observed behavior rather than merely recognizing the behavior itself.

4.1 Structural Concepts

Structural concepts describe the relational conditions that govern the trajectories available to a system.

Relationships

Relationships describe how variables influence one another.

Variables are measurements.

Relationships are patterns of interaction among measurements.

A variable considered in isolation may provide limited information about system behavior. Its significance often depends on how it relates to other variables within the system.

Relationships may express:

- Coupling
- Dependency
- Competition
- Compensation
- Coordination
- Feedback
- Constraint

Within Constraint Geometry, relationships constitute the primary substrate from which higher-order structures emerge.

Constraints arise from relationships.

Dominance emerges among competing constraints.

Trajectories emerge from the interaction of those constraints through time.

For this reason, the framework treats relationships as the fundamental unit of structural organization rather than isolated variables alone.

Constraints

A constraint is any condition that alters the set of trajectories available to a system.

Constraints may enable, restrict, redirect, or stabilize behavior. They are defined not by their physical form, but by their influence on system evolution.

Importantly, constraints emerge from relationships rather than existing independently of them.

Variables are measurements.

Relationships describe interactions.

Constraints are the conditions created by those interactions that shape future behavior.

Examples include:

- Resource limitations
- Capacity limits
- Information bottlenecks
- Regulatory dependencies
- Timing restrictions
- Environmental pressures

Within Constraint Geometry, system behavior is understood primarily through interactions among constraints rather than through isolated variables.

The benchmark question is therefore not simply:

Which variable changed?

but:

Which constraints govern the trajectory of the system?

Dominance

Dominance refers to the relative influence of a constraint on system trajectory.

A constraint is dominant when modifications to that constraint would alter the trajectory more substantially than modifications to competing constraints.

Dominance is not fixed. As conditions evolve, a previously secondary constraint may become dominant, while compensation may suppress one constraint and amplify another.

Identifying dominant constraints is therefore a central mechanism-attribution task.

Alignment

Alignment refers to the degree of correspondence between system responses and the constraints that actually govern behavior.

A system may possess information, resources, and adaptive capacity yet remain ineffective if those capabilities are directed toward the wrong problem.

Alignment therefore concerns whether available responses are matched to the constraints that matter most.

High alignment increases the likelihood that available capacity influences the conditions most responsible for system behavior.

Low alignment reduces effectiveness even when substantial resources remain available.

4.6 Structural Alignment

Structural alignment refers to the degree of correspondence between an inferred structural explanation and the generative structure responsible for an observed trajectory.

Structural alignment is conceptually analogous to predictive accuracy, but applies to explanations rather than outcomes. Where predictive accuracy measures whether a model identifies the correct outcome, structural alignment measures whether it identifies the correct structural explanation.

Within benchmark environments, the true generative structure is known by construction. This makes it possible to compare a model's inferred explanation against the structure that actually produced the observed behavior.

Structural alignment forms the conceptual foundation of mechanism-attribution evaluation. The primary metric derived from this concept is Structural Alignment Accuracy (SAA), which measures the extent to which a model's inferred structural explanation corresponds to the generative structure underlying an observed trajectory.

Because benchmark environments expose both outcomes and generating structures, SAA enables direct comparison between outcome prediction performance and mechanism-attribution performance on the same tasks.

The precise operational definition of SAA depends on the benchmark layer and evaluation setting and is introduced in the Evaluation Metrics section.

This distinction provides the foundation for the evaluation methodology developed in the remainder of the paper.

5. The Stability Geometry Model

The preceding sections introduced the core concepts of Constraint Geometry individually. This section integrates those concepts into a unified structural model describing how trajectories emerge, persist, adapt, and fail.

The Stability Geometry Model serves as the organizing architecture of the benchmark framework.

It is not intended as a biological theory, a clinical model, or a description of any specific real-world system. Rather, it provides a structural abstraction used to construct benchmark environments, generate trajectories, and define mechanism-attribution targets.

The model begins with a simple observation:

Systems do not move directly from information to outcomes.

Information must first become available, be interpreted correctly, aligned with the constraints governing behavior, translated into effective action, and supported by compensatory processes before stability can emerge.

Failures may occur at any stage in this process.

Consequently, similar outcomes may arise from fundamentally different structural conditions.

The purpose of the Stability Geometry Model is to make those conditions visible and distinguishable.

5.1 Conceptual Architecture

At its highest level, the model consists of five interacting layers that shape system trajectories:

Signal

↓

Readability

↓

Alignment

↓

Action

↓

Compensation

↓

Trajectory

↓

Stability

The first five layers describe the mechanisms through which information influences behavior.

Trajectory represents the evolving path produced by their interaction.

Stability emerges from that trajectory rather than existing as an independent layer.

The model therefore does not treat stability as a primitive property of a system. Instead, stability is understood as the consequence of how information is interpreted, how constraints are identified, how responses are deployed, and how stress is absorbed through time.

The benchmark objective is not merely to recognize stability or instability.

It is to determine which layer or combination of layers best explains the observed trajectory.

Recursive Dynamics

Although presented sequentially, the architecture is not strictly linear.

Actions alter future signals, compensation changes future constraints, and trajectories generate new information that feeds back into subsequent cycles of interpretation and response.

A successful intervention may improve future readability by reducing noise. Compensation may temporarily suppress a dominant constraint, changing which signals become most relevant. Likewise, evolving trajectories may reveal information that was previously hidden or make previously important information irrelevant.

The layers of the architecture therefore influence one another across time rather than operating as isolated stages.

For this reason, the Stability Geometry Model should be understood as a recursive process rather than a one-way pipeline. The linear representation is intended to illustrate the primary flow of influence, while the underlying dynamics remain iterative, adaptive, and feedback-driven.

Constraints Across the Architecture

Although the Stability Geometry Model is presented as a sequence of interacting layers, constraints operate across the entire architecture rather than occupying a single layer within it.

Constraints determine which signals become relevant, which interpretations are meaningful, which alignments are possible, which actions are available, and which trajectories remain accessible.

In this sense, constraints do not sit inside the architecture.

They shape the architecture.

The Stability Geometry Model can therefore be understood as a framework for analyzing how information, action, and adaptation unfold within a landscape defined by interacting constraints.

Mechanism attribution is ultimately concerned with identifying the constraint structure most responsible for producing an observed trajectory.

5.2 Signal

Signal represents information available to the system that may influence behavior.

Signals may originate from internal conditions, environmental changes, resource availability, emerging threats, or changes in system state.

Signal answers a simple question:

What information is available to the system?

A signal need not be complete, accurate, or interpretable.

Its availability alone does not guarantee that it influences behavior.

Within the Stability Geometry Model, signal serves as the entry point through which information enters the system.

Without signal, adaptation is impossible because the system lacks awareness of conditions requiring response.

5.3 Readability

Readability concerns whether available signals can be interpreted correctly.

Signal availability and signal usability are not the same thing.

Information may exist without being understandable.

Relevant signals may be obscured by noise, degradation, ambiguity, or competing information.

Readability answers:

Can the system correctly interpret the signal?

Failures at this layer produce situations in which information is available but cannot be transformed into understanding.

Such failures may generate instability despite adequate information availability.

The distinction between signal absence and signal unreadability forms one of the central mechanism-attribution challenges within the framework.

5.4 Alignment

Alignment concerns whether interpreted information corresponds to the constraints that actually govern system behavior.

A system may correctly understand available information yet focus on the wrong problem.

Resources may be directed toward secondary constraints while dominant constraints remain unaddressed.

Alignment answers:

Is the system focused on the right problem?

Failures at this layer generate coherent but misdirected behavior.

The system acts rationally relative to its interpretation of events, yet remains ineffective because its responses target the wrong constraints.

5.5 Action

Action represents the capacity to translate aligned understanding into effective response.

Readability without action produces awareness without adaptation.

Alignment without action produces recognition without change.

Action answers:

Can the system respond?

Failures at this layer occur when a system correctly identifies relevant constraints but lacks the resources, authority, coordination, timing, or capacity required to act effectively.

A system may therefore understand a problem perfectly and still fail because it cannot alter the conditions responsible for that problem.

5.6 Compensation

Compensation refers to processes that preserve function despite unresolved constraints.

Compensation occupies a unique position within the model.

Unlike the preceding layers, compensation does not determine whether a problem is recognized or addressed. Instead, it determines how long the system can remain functional despite unresolved structural pressures.

Compensation answers:

Can the system absorb stress while maintaining function?

Compensatory processes can conceal failures occurring elsewhere in the architecture.

A system may therefore appear stable even when:

- Signals are degrading
- Readability is declining
- Alignment is deteriorating
- Actions are ineffective

As long as compensatory capacity remains sufficient, observable outcomes may remain relatively unchanged.

This property makes compensation one of the primary sources of ambiguity within mechanism-attribution tasks.

5.7 Trajectory

Trajectory is the central integrating construct of the Stability Geometry Model.

It represents the path a system follows through time as information is interpreted, constraints are addressed, actions are executed, and compensatory processes operate.

Trajectory answers:

How is the system evolving?

Outcomes describe endpoints.

Trajectories describe the processes that produce those endpoints.

This distinction is fundamental because similar outcomes may emerge from different trajectories, while similar trajectories may diverge as conditions change.

Mechanism attribution therefore concerns trajectories rather than outcomes alone.

The central question becomes:

What sequence of structural conditions produced the observed behavior?

5.8 Stability

Stability is not an independent layer within the architecture.

It is an emergent property of the trajectory generated by the preceding layers.

Stability answers:

Can the system sustain coherent behavior through time?

Stable systems are not necessarily free of stress, uncertainty, or constraint.

They are systems in which information remains sufficiently interpretable, responses remain sufficiently aligned, actions remain sufficiently effective, and compensatory capacity remains sufficient to preserve function.

Instability emerges when structural pressures accumulate beyond the system's ability to absorb or adapt to them.

Within the Stability Geometry Model, stability is therefore treated as a consequence rather than a cause.

5.9 Structural Failure Modes

One of the central insights of the Stability Geometry Model is that similar outcomes may originate from different structural failures.

Signal Failure

Relevant information is absent, unavailable, or inaccessible.

The system cannot respond because critical information never enters the architecture.

Readability Failure

Information is available but cannot be interpreted correctly.

The signal is present, but its meaning cannot be extracted.

Alignment Failure

Information is understood but directed toward the wrong constraint.

The system focuses on the wrong problem.

Action Failure

The correct problem is identified, but effective response is impossible.

Understanding exists without sufficient capacity to act.

Compensation Failure

Structural pressures exceed the system's buffering capacity.

Previously hidden instability becomes visible.

Stability Failure

Accumulated structural failures exceed the system's ability to maintain coherent behavior.

Observable breakdown occurs.

The observable outcome may be similar in each case.

The governing mechanism is not.

5.10 Structural Interpretation

The Stability Geometry Model provides a framework for interpreting outcomes structurally rather than descriptively.

Traditional prediction asks:

What happened?

The Stability Geometry Model asks:

Why did it happen?

More specifically:

Where within the architecture did failure originate?

For example, two systems may exhibit identical instability.

One may fail because relevant signals never entered the system.

The other may fail because compensatory capacity was exhausted.

The outcome appears similar.

The structural explanation differs.

This distinction lies at the heart of mechanism attribution.

Under the Stability Geometry Model, explanations become layered rather than categorical.

A system is not simply stable or unstable.

It is stable or unstable for specific structural reasons.

5.11 Why Geometry?

The term *geometry* refers to the configuration of relationships and constraints that shapes the space of available trajectories.

Just as physical geometry constrains movement through space, constraint geometry constrains movement through state space.

A system's future behavior is not determined solely by its current state. It is also shaped by the structural configuration of constraints acting upon that state.

Different constraint configurations may produce similar outcomes while making different trajectories available. Likewise, similar configurations may produce different outcomes as dominance relationships, compensation, and observability change through time.

The objective of Constraint Geometry is therefore not merely to predict where a system will end up, but to identify the structural configuration governing how it moves.

Benchmark tasks are designed to evaluate whether models can correctly attribute observed trajectories to the constraint structures that produce them.

5.12 Benchmark Implications

The Stability Geometry Model provides the conceptual foundation for benchmark construction.

Rather than evaluating whether a model predicts an outcome, benchmark tasks can evaluate whether a model correctly identifies the structural conditions responsible for that outcome.

Examples include:

Signal Attribution Tasks

Can the model identify missing, degraded, or inaccessible signals?

Readability Tasks

Can the model distinguish between signal absence and signal misinterpretation?

Alignment Tasks

Can the model identify when resources are directed toward secondary rather than dominant constraints?

Action Tasks

Can the model distinguish between inability to act and inability to understand?

Compensation Tasks

Can the model recognize hidden instability masked by compensatory processes?

Stability Tasks

Can the model identify the dominant structural source of instability?

Together, these tasks transform mechanism attribution into a measurable benchmark objective.

The Stability Geometry Model therefore serves as the bridge between the conceptual framework introduced in earlier sections and the benchmark architectures introduced later in the paper.

More fundamentally, it provides a systematic way to ask not merely whether a system succeeds or fails, but why it succeeds or fails.

That distinction lies at the center of the Constraint Geometry framework.

6. Why Existing Benchmarks Are Easy to Shortcut

A benchmark measures only the capabilities its design makes necessary.

This observation appears simple, yet it has profound implications for evaluation. Benchmark performance is often interpreted as evidence of reasoning, understanding, or capability. In

practice, however, high benchmark scores may arise from many different strategies, not all of which correspond to the capability a benchmark was intended to measure.

A model may succeed because it has learned the target capability.

A model may also succeed because the benchmark permits simpler alternatives.

These alternatives take many forms: label leakage, proxy variables, shortcut learning, hidden determinism, spurious correlations, and dataset-construction artifacts. Although they differ in appearance, they share a common property: they allow benchmark success without requiring the reasoning process the benchmark was designed to evaluate.

The problem is not that models exploit such opportunities. From the perspective of optimization, doing so is entirely rational.

The problem is interpretability.

If a benchmark can be solved without exercising the capability it claims to measure, benchmark performance becomes difficult to interpret. High scores may reveal that a model has found an efficient strategy rather than demonstrating that it possesses the capability the benchmark was intended to assess.

This challenge is particularly important for mechanism-attribution tasks. A benchmark intended to evaluate structural reasoning must ensure that outcomes cannot be inferred through simpler predictive strategies. Otherwise, high performance may reflect prediction rather than mechanism attribution.

The purpose of this section is not to criticize existing benchmarks. Rather, it is to identify a family of benchmark failure modes that emerge when the intended capability can be substituted by a simpler alternative.

These failure modes motivate the benchmark-design principles introduced in the following chapter.

6.1 Capability Substitution

Although benchmark failure modes appear diverse, they often share a common structure.

In each case, a benchmark is intended to evaluate one capability while permitting success through another.

A benchmark intended to measure reasoning may reward correlation.

A benchmark intended to measure structure may reward prediction.

A benchmark intended to measure mechanism attribution may reward proxy detection.

The result is what can be described as **capability substitution**: the replacement of the intended reasoning process with a simpler strategy that produces similar benchmark performance.

Capability substitution is not a flaw in the model.

It is a flaw in the benchmark.

The central challenge of benchmark design is therefore not merely increasing difficulty. It is ensuring that success requires the capability the benchmark claims to evaluate.

The remaining sections describe common forms of capability substitution that arise in practice.

6.2 Label Leakage

Label leakage occurs when information used to construct the target label becomes directly or indirectly available to the model.

In such situations, the benchmark ceases to evaluate reasoning and instead evaluates the model's ability to detect information already encoding the answer.

Examples include:

- Features derived from the target variable
- Variables recorded after the outcome occurred
- Metadata correlated with labels
- Dataset-construction artifacts

Label leakage is particularly problematic because it often produces extremely high benchmark performance.

The resulting scores may appear impressive while providing little evidence that the intended capability has actually been measured.

Within mechanism-attribution benchmarks, leakage can occur when structural labels become directly recoverable from observable variables.

A benchmark intended to evaluate mechanism attribution must therefore ensure that structural explanations cannot be reconstructed from a single leaked feature.

6.3 Single-Variable Separation

A benchmark exhibits single-variable separation when one variable alone reliably determines the correct answer.

Under such conditions, the benchmark may appear complex while requiring only threshold detection.

For example:

- Values above a threshold correspond to one class
- Values below a threshold correspond to another class

A model can achieve high performance without reasoning about relationships, interactions, compensation, dominance, or competing mechanisms.

Single-variable separation often emerges unintentionally during dataset construction. When classes become cleanly separable along a single dimension, the benchmark no longer evaluates the intended structure.

For mechanism-attribution tasks, this failure mode is especially problematic because it bypasses the need to reason about relationships among constraints.

A benchmark intended to measure structural reasoning must therefore prevent any single variable from becoming sufficient for correct classification.

6.4 Proxy Prediction

Proxy prediction occurs when a variable correlates strongly with the target label despite not representing the mechanism the benchmark intends to measure.

The proxy functions as a shortcut.

A model learns to predict the proxy rather than reason about the underlying structure.

Examples include:

- Age serving as a proxy for disease risk
- Resource consumption serving as a proxy for system failure
- Aggregate scores serving as proxies for underlying mechanisms

Proxy prediction can be difficult to detect because benchmark performance may remain high while the intended reasoning process is absent.

The challenge is not that proxies are useless.

The challenge is that benchmark success becomes ambiguous.

It becomes unclear whether the model has identified the mechanism or merely discovered a correlated indicator.

Constraint Geometry seeks to minimize proxy prediction by ensuring that multiple mechanisms remain consistent with similar observable outcomes.

6.5 Shortcut Learning

Shortcut learning refers to the broader phenomenon in which models exploit statistical regularities that differ from the reasoning process benchmark designers intended to evaluate.

This behavior has been documented across many machine-learning domains.

Models routinely discover solutions that maximize benchmark performance while minimizing the amount of reasoning required.

Examples include:

- Exploiting annotation artifacts
- Detecting dataset-specific biases
- Learning spurious correlations
- Memorizing construction patterns

From the perspective of optimization, shortcut learning is entirely rational.

The model is solving the task as presented.

The problem lies in benchmark interpretation.

A benchmark intended to evaluate structural reasoning becomes unreliable if shortcut strategies consistently outperform structural analysis.

Constraint Geometry therefore treats shortcut resistance as a primary design objective rather than an optional refinement.

6.6 Hidden Determinism

Hidden determinism occurs when dataset construction creates deterministic mappings that are not immediately obvious to benchmark designers.

A benchmark may appear structurally rich while actually being governed by a hidden rule.

Examples include:

- One variable always dominates above a threshold
- One interaction always determines the label
- One latent factor perfectly predicts the outcome

The resulting dataset may contain many variables and many relationships while remaining reducible to a simple deterministic rule.

This problem is especially relevant in synthetic benchmark construction.

Synthetic environments provide greater control than real-world data, but they also create opportunities for unintended determinism.

Mechanism-attribution benchmarks must therefore be explicitly tested for hidden separability and latent shortcut structures.

Otherwise, apparent structural complexity may conceal a fundamentally simple task.

6.7 Equifinality Collapse

A central motivation for Constraint Geometry is the observation that different mechanisms can produce similar outcomes.

This property is often referred to as equifinality.

Many benchmark datasets collapse such cases into a single outcome label. When this occurs, information about the underlying mechanism is discarded during dataset construction.

The resulting benchmark may remain useful for prediction while becoming incapable of evaluating mechanism attribution.

From the perspective of structural reasoning, this represents a significant loss of information.

If multiple mechanisms produce the same outcome, a benchmark intended to evaluate mechanism attribution must preserve those distinctions rather than eliminate them.

Constraint Geometry therefore treats mechanistic diversity among similar outcomes as a design requirement rather than a nuisance variable.

6.8 Outcome Without Mechanism

Many benchmark tasks evaluate outcomes without evaluating the explanations that produce them.

This design is entirely appropriate when outcome prediction is the primary objective.

The limitation emerges when benchmark performance is interpreted as evidence of understanding.

A model may correctly predict:

- Disease progression
- System failure
- Recovery
- Risk level

while remaining unable to explain the mechanism responsible for that prediction.

In such cases, benchmark success demonstrates outcome recognition.

It does not necessarily demonstrate mechanism attribution.

This distinction forms one of the central motivations for the Constraint Geometry framework.

Outcome prediction and mechanism attribution may overlap.

They are not necessarily equivalent.

6.9 Correlation Without Structure

Correlation is often sufficient for successful prediction.

Structure is often required for explanation.

The distinction matters because correlated variables do not necessarily reveal the processes responsible for observed behavior.

Two variables may correlate because:

- One influences the other
- Both share a common cause
- Compensation links them indirectly
- System dynamics create recurring patterns

Prediction may succeed without distinguishing among these possibilities.

Mechanism attribution cannot.

A benchmark intended to evaluate structural reasoning must therefore require models to discriminate among competing explanations rather than merely identify statistical associations.

Constraint Geometry is built around this principle.

The objective is not simply to detect patterns.

It is to evaluate whether models can identify the structural explanations most consistent with those patterns.

6.10 Why High Accuracy Can Be Misleading

High benchmark accuracy is often interpreted as evidence that a model has mastered the capability being evaluated.

This interpretation is not always justified.

A model may achieve strong performance because:

- Labels leak into features
- A single variable separates classes
- Proxy variables encode the answer
- Shortcut strategies exist
- Hidden deterministic rules dominate the dataset
- Mechanistic diversity has been collapsed into outcome labels
- Correlations substitute for structural understanding

In each case, benchmark performance remains high while confidence in what has actually been measured declines.

This observation becomes especially important for mechanism-attribution benchmarks.

If a model can achieve high performance without reasoning about constraints, trajectories, compensation, dominance, or structural explanations, then benchmark scores provide limited evidence that those capabilities are present.

The challenge for benchmark design is therefore not simply to maximize difficulty.

It is to ensure that success requires the capability the benchmark claims to evaluate.

Constraint Geometry is built around this principle.

The design rules introduced in the following chapter are intended to make mechanism attribution difficult to shortcut, difficult to reduce to simple prediction, and difficult to solve without engaging the structural reasoning processes the benchmark is designed to measure.

7. Constraint Geometry Design Rules

The previous section identified a family of benchmark failure modes that can undermine interpretation of benchmark results. Label leakage, proxy prediction, shortcut learning, hidden determinism, equifinality collapse, and capability substitution all create situations in which benchmark success may occur without exercising the capability a benchmark claims to evaluate.

Constraint Geometry is built around a simple principle:

A benchmark intended to evaluate mechanism attribution should require mechanism attribution for successful performance.

This principle appears straightforward, but it has significant implications for dataset construction.

A benchmark cannot merely contain structural complexity.

It must require engagement with that complexity.

If simpler predictive strategies remain sufficient, the benchmark risks evaluating prediction, correlation detection, threshold recognition, or proxy identification rather than structural reasoning.

The rules introduced in this section should not be understood as independent heuristics. They are design constraints derived from a common objective:

Preserve mechanism-attribution signal while suppressing capability substitution.

Collectively, these rules seek to preserve structural ambiguity, maintain mechanistic diversity, and prevent benchmark performance from collapsing into simpler forms of prediction.

Together they define the design philosophy underlying the Constraint Geometry Benchmark Suite.

7.1 Structural Ambiguity

A central concept underlying the framework is **structural ambiguity**.

Structural ambiguity exists whenever multiple structural explanations remain consistent with the available evidence.

In many benchmark settings, ambiguity is treated as a problem to be eliminated. Constraint Geometry treats it as a design requirement.

Mechanism attribution becomes meaningful only when multiple explanations remain plausible.

If observations uniquely determine the correct explanation, attribution becomes trivial. If outcomes uniquely determine mechanisms, attribution collapses into prediction.

Structural ambiguity arises whenever multiple structural explanations remain consistent with the available evidence.

Equifinality represents one important source of such ambiguity, but it is not the only one.

Ambiguity may also arise through compensation, partial observability, contested dominance relationships, delayed effects, signal degradation, incomplete trajectories, or competing constraints.

In each case, multiple explanations remain plausible despite access to the same observations.

Constraint Geometry deliberately preserves this ambiguity because mechanism attribution becomes meaningful only when alternative structural explanations remain viable.

7.2 Meta-Rule: Mechanism Attribution Must Be Separable from Outcome Prediction

This principle defines the central requirement of the Constraint Geometry framework.

A benchmark cannot evaluate mechanism attribution if mechanisms can be recovered directly from outcomes.

When outcome labels uniquely determine mechanism labels, mechanism attribution collapses into prediction.

For this reason, outcome recognition and mechanism attribution must remain partially independent benchmark targets.

Constraint Geometry preserves two complementary forms of structural ambiguity.

Many-to-One Mapping

Different mechanisms → same outcome

Multiple structural explanations may converge on the same observable state.

Failure, instability, recovery, adaptation, or degradation may arise through different combinations of constraints, compensation dynamics, dominance relationships, or information failures.

If outcomes uniquely identify mechanisms, attribution becomes trivial.

One-to-Many Mapping

Same mechanism → different outcomes

A single structural mechanism may produce different outcomes depending on context, timing, compensation, observability, intervention history, or competing constraints.

If mechanisms uniquely determine outcomes, mechanisms become hidden outcome labels rather than structural explanations.

Together, these mappings prevent mechanism attribution from collapsing into outcome prediction.

The remaining design rules can be understood as specific mechanisms for preserving these forms of structural ambiguity and maintaining the separation between outcome recognition and mechanism attribution.

Structural Diversity Rules

Structural diversity prevents benchmark tasks from collapsing into deterministic mappings between observations and labels.

Rule 1: No Single Variable Should Determine the Label

No individual variable should be sufficient to determine the correct answer.

If a single measurement consistently separates classes, the benchmark becomes reducible to threshold detection rather than structural reasoning.

Constraint Geometry therefore requires labels to emerge from relationships among variables rather than from individual variables considered independently.

The benchmark question should not be:

Which variable crossed a threshold?

It should be:

Which structural configuration best explains the observations?

Rule 2: Different Mechanisms Must Sometimes Produce Identical Outcomes

This rule operationalizes the Many-to-One Mapping principle.

Multiple structural explanations should occasionally generate the same observable outcome.

If every outcome uniquely identifies its mechanism, mechanism attribution becomes trivial.

Constraint Geometry therefore preserves mechanistic diversity among similar outcomes.

The outcome may be identical.

The explanation is not.

Rule 3: Identical Mechanisms Must Sometimes Produce Different Outcomes

This rule operationalizes the One-to-Many Mapping principle.

The same structural mechanism should not always generate the same outcome.

Context matters.

Compensation may succeed in one case and fail in another.

Timing may alter the effect of an intervention.

Different dominance relationships may produce divergent trajectories despite similar underlying mechanisms.

Without such cases, mechanisms become deterministic labels.

Constraint Geometry therefore includes situations in which identical mechanisms remain consistent with multiple outcomes.

Stability–Burden Decoupling Rules

These rules prevent severity, burden, or visible instability from becoming proxies for structural explanation.

Rule 4: High Burden Must Sometimes Remain Stable

High apparent burden should not always imply instability.

In many real systems, substantial stress can be absorbed through compensation, redundancy, adaptation, or reserve capacity.

Constraint Geometry therefore includes cases in which systems remain stable despite substantial burden.

These cases force models to evaluate the structural conditions that preserve stability rather than relying on burden alone.

Rule 5: Low Burden Must Sometimes Fail

The inverse condition must also exist.

Low apparent burden should not always imply stability.

Systems may fail despite modest observable stress when critical constraints become dominant, signals become unreadable, compensatory capacity collapses, or key dependencies are disrupted.

Constraint Geometry intentionally violates the assumption that more burden necessarily implies more failure.

Failure must sometimes emerge under conditions that appear relatively benign.

Rule 6: Observed Stability Must Sometimes Conceal Instability

Apparent stability can be misleading.

Compensation may conceal growing structural pressure.

Reserve capacity may delay visible failure.

Signals may remain within expected ranges despite increasing fragility.

Constraint Geometry therefore includes situations in which observable stability masks underlying instability.

The benchmark objective becomes identifying structural conditions that remain hidden from surface observations.

Constraint Dynamics Rules

These rules ensure that trajectories remain shaped by interacting constraints rather than reducible to deterministic relationships.

Rule 7: Compensation Must Sometimes Override Dominant Signals

Compensation is one of the primary mechanisms through which structural ambiguity emerges.

Observable signals may suggest instability while compensatory processes preserve function.

Conversely, systems may appear healthy despite accumulating hidden instability.

Constraint Geometry therefore requires compensation to occasionally override what would otherwise appear to be dominant indicators.

These cases create situations in which outcome recognition alone becomes insufficient.

Rule 8: Dominance Must Sometimes Be Ambiguous

Many real systems contain multiple active constraints simultaneously.

Not all situations contain a clearly dominant mechanism.

Several constraints may exert comparable influence over the trajectory of a system.

Constraint Geometry therefore includes situations in which dominance relationships are contested, uncertain, or evolving.

The objective becomes identifying the most plausible governing constraint under uncertainty.

Information Constraint Rules

These rules prevent information availability from becoming a shortcut to structural explanation.

Rule 9: Partial Observability Must Sometimes Remain Stable

Incomplete information should not always imply instability.

Many systems continue to function effectively despite missing observations.

Compensation, redundancy, robustness, and adaptive behavior can preserve stability even when critical information is unavailable.

If missing information always predicts failure, partial observability becomes a proxy for instability.

Constraint Geometry therefore includes stable systems operating under incomplete information.

These cases require models to distinguish between uncertainty and failure.

The absence of information should not automatically imply structural collapse.

Design Logic

The purpose of these rules is not to make benchmarks harder.

Difficulty is not evidence of validity.

A benchmark may be extremely difficult and still fail to evaluate the capability it claims to measure. Conversely, a relatively simple benchmark may provide strong evidence that a capability is present.

The purpose of these rules is to preserve mechanism-attribution signal while suppressing capability substitution.

The central design question is therefore not:

How difficult is the benchmark?

but:

What capability does benchmark success require?

Foundational Principles

Principle	Function
Structural Ambiguity	Preserves multiple plausible structural explanation
Mechanism Attribution Must Be Separable from Outcome Prediction	Preserves mechanism-attribution signal

Implementation Rules

Design Rule	Primary Failure Mode Addressed
No Single Variable Should Determine the Label	Single-variable separation
Different Mechanisms Must Sometimes Produce Identical Outcomes	Equifinality collapse
Identical Mechanisms Must Sometimes Produce Different Outcomes	Hidden determinism
High Burden Must Sometimes Remain Stable	Proxy prediction
Low Burden Must Sometimes Fail	Proxy prediction
Observed Stability Must Sometimes Conceal Instability	Correlation without structure
Compensation Must Sometimes Override Dominant Signals	Shortcut learning
Dominance Must Sometimes Be Ambiguous	Hidden determinism
Partial Observability Must Sometimes Remain Stable	Proxy prediction

The stronger these controls become, the more confidently benchmark performance can be interpreted as evidence of mechanism attribution rather than prediction, correlation detection, proxy exploitation, or shortcut learning.

In this sense, the design rules function as the benchmark analogue of experimental controls. Their purpose is not to guarantee perfect measurement, but to reduce alternative explanations for benchmark success.

Constraint Geometry is therefore designed so that structural explanations cannot be reliably inferred through simple thresholds, proxy variables, outcome labels, or deterministic shortcuts.

Success requires engagement with constraints, compensation, dominance, trajectories, and competing structural explanations.

The next section translates these principles into a concrete benchmark architecture and evaluation framework.

8. Why Mechanism Attribution Matters

The preceding sections argued that mechanism attribution may represent a distinct benchmark target and introduced a framework for constructing benchmark tasks that preserve mechanism-attribution signal.

A natural question follows:

Why does mechanism attribution matter?

If outcome prediction is accurate, why should benchmark evaluation concern itself with structural explanations at all?

The answer depends on what benchmark performance is intended to represent.

If the objective is prediction alone, outcome accuracy may be sufficient.

If the objective includes explanation, intervention, adaptation, failure analysis, planning, or decision support, then identifying the mechanisms responsible for observed outcomes becomes increasingly important.

Constraint Geometry is motivated by a simple observation:

Outcomes describe what happened. Mechanisms explain why it happened.

These questions are related.

They are not the same question.

The purpose of this section is not to argue that prediction is unimportant. Prediction remains one of the most valuable capabilities in artificial intelligence.

Rather, the goal is to examine whether outcome prediction and mechanism attribution provide different forms of information about system behavior—and whether both may be worth measuring.

8.1 Prediction and Explanation Answer Different Questions

Prediction answers a specific question:

What outcome is likely to occur?

Explanation answers a different question:

What structural conditions are producing that outcome?

These questions often overlap.

A model that correctly identifies a governing mechanism may improve its ability to predict future outcomes.

Likewise, a model that predicts outcomes accurately may sometimes infer the underlying mechanism.

The overlap, however, is not complete.

A model may successfully predict:

- Failure
- Recovery
- Disease progression
- System instability
- Resource depletion

without identifying the structural explanation responsible for those predictions.

Similarly, a model may identify the dominant mechanism governing a trajectory even when future outcomes remain uncertain.

Prediction and explanation therefore represent different informational targets.

Constraint Geometry investigates whether benchmark evaluation can assess those targets independently.

8.2 Outcomes Compress Information

One way to understand the distinction is through information compression.

Outcomes summarize trajectories.

They collapse many possible structural histories into a smaller set of observable states.

This compression is useful.

It allows benchmark tasks to focus on clinically relevant events, operational outcomes, failures, recoveries, and other meaningful endpoints.

At the same time, information is lost.

Multiple trajectories may converge on the same outcome.

Multiple structural explanations may remain consistent with the same observation.

This phenomenon was introduced earlier as **Many-to-One Mapping**:

Different mechanisms → same outcome

When this occurs, outcome recognition alone cannot determine which mechanism generated the outcome.

Mechanism attribution attempts to recover that missing structural information.

The objective is not to replace prediction.

It is to distinguish among explanations that prediction alone cannot resolve.

8.3 Explanations Preserve Information

If outcomes compress information, explanations preserve it.

Structural explanations retain distinctions that outcomes eliminate.

While outcomes summarize trajectories into observable states, explanations preserve information about the pathways, constraints, relationships, and mechanisms that generated those states.

This distinction becomes particularly important when multiple trajectories converge on the same outcome.

The outcome remains identical.

Information about how that outcome emerged does not.

Mechanism attribution can therefore be understood as an attempt to recover information lost during outcome compression.

Within the Constraint Geometry framework, structural explanations serve as representations of the generative conditions responsible for an observed trajectory.

Their purpose is not merely descriptive.

They preserve distinctions among alternative structural explanations that would otherwise become indistinguishable at the level of outcomes alone.

Outcome recognition identifies where a system arrived.

Mechanism attribution attempts to identify how it arrived there.

8.4 Structural Explanation Matters

Within Constraint Geometry, explanation is not treated as a post-hoc narrative.

It is treated as an attribution task.

The objective is to determine which structural explanation remains most consistent with the available evidence.

This distinction matters because different explanations may remain compatible with the same outcome.

For example:

- Compensation collapse may produce failure.
- Signal unreadability may produce failure.
- Dominance shifts may produce failure.
- Alignment breakdown may produce failure.

The outcome is identical.

The explanations are not.

A benchmark that evaluates only outcome prediction treats these cases as equivalent.

A benchmark that evaluates mechanism attribution attempts to distinguish among them.

This distinction forms the conceptual core of the framework.

8.5 Intervention Operates on Mechanisms

The practical importance of mechanism attribution becomes most visible when interventions are considered.

Interventions do not operate on outcomes.

They operate on mechanisms.

A system does not become stable because instability has been recognized.

It becomes stable because the constraints responsible for instability have been altered.

This creates an important asymmetry.

Similar outcomes may require different interventions.

Different interventions may produce different results even when applied to systems with similar observable states.

Consequently, intervention effectiveness often depends less on recognizing the outcome than on identifying the structural explanation responsible for that outcome.

This observation does not imply that intervention always requires perfect mechanism attribution.

Rather, it suggests that explanation and intervention are naturally linked in ways that prediction alone may not capture.

8.6 Failure Reconstruction

Mechanism attribution can also be understood as a form of structural reconstruction.

Outcome prediction asks:

What happened?

Mechanism attribution asks:

What produced what happened?

This distinction appears across many disciplines.

Aviation investigations rarely stop after determining that an aircraft crashed.

Engineering investigations rarely stop after determining that a bridge failed.

Operational reviews rarely stop after determining that a system became unstable.

Instead, investigators attempt to reconstruct the sequence of structural conditions responsible for the outcome.

Constraint Geometry treats mechanism attribution as a benchmark analogue of this process.

The objective is not merely to identify the endpoint of a trajectory but to reconstruct the structural conditions that generated it.

8.7 Governing Constraint Identification

At the center of the framework lies a specific attribution problem:

Which constraint governs the observed trajectory?

Most complex systems contain multiple active constraints simultaneously.

Some exert substantial influence.

Others are secondary.

Not all contribute equally to future behavior.

Mechanism attribution therefore requires more than identifying constraints.

It requires identifying which constraints matter most.

This task is referred to within the framework as **governing constraint identification**.

The objective is to determine which constraint or combination of constraints most strongly shapes the trajectory of the system.

Governing constraint identification represents the benchmark-level operationalization of mechanism attribution.

Rather than asking whether a model can merely recognize an outcome, the benchmark asks whether it can identify the structural conditions most responsible for producing that outcome.

This concept connects directly to dominance, compensation, alignment, latency, and partial observability.

Together these concepts determine which structural explanation is most consistent with the available evidence.

8.8 Why Mechanism Attribution Represents a Distinct Benchmark Target

The central claim of this paper is not that mechanism attribution is necessarily a distinct cognitive capability.

That remains an empirical question.

The claim is more modest:

Mechanism attribution represents a distinct benchmark target.

The distinction arises because successful outcome prediction does not guarantee successful mechanism attribution.

A model may correctly identify an outcome while failing to identify the structural explanation responsible for it.

Likewise, a model may correctly identify the governing mechanism even when outcome prediction remains uncertain.

If such situations occur, then outcome prediction and mechanism attribution cannot be assumed to measure the same thing.

Constraint Geometry therefore proposes evaluating them separately.

This proposal motivates the dual-target architecture introduced later in the paper:

- **Target A:** Outcome Recognition
- **Target B:** Mechanism Attribution

The relationship between these targets becomes an empirical question rather than an assumption.

One possible outcome is that the targets prove highly correlated.

Another is that meaningful separation emerges.

The benchmark framework is designed to investigate that question directly.

8.9 Illustrative Example

The distinction can be illustrated through a simplified example.

Case A

Outcome: Failure

Mechanism: Repair Capacity Collapse

Adaptive capacity has been exhausted.

The system can no longer maintain structural integrity despite ongoing demand.

A successful intervention would focus on restoring or augmenting repair capacity.

Case B

Outcome: Failure

Mechanism: Signal Readability Collapse

Relevant information no longer reaches the processes responsible for adaptation.

Repair capacity may remain available, but the system cannot coordinate its use effectively.

A successful intervention would focus on restoring signal transmission and interpretability.

Comparison

Feature	Case A	Case B
Outcome	Failure	Failure
Governing Mechanism	Repair Capacity Collapse	Signal Readability Collapse
Dominant Constraint	Capacity Exhaustion	Information Failure
Plausible Intervention	Restore Capacity	Restore Readability

The observable outcome is identical.

The governing structures are not.

The interventions are not.

A benchmark that evaluates only outcome prediction treats these cases as equivalent.

A benchmark that evaluates mechanism attribution distinguishes between them.

This distinction captures the central motivation of the Constraint Geometry framework.

The question is not merely whether a model can identify what happened.

The question is whether it can identify the structural explanation responsible for what happened.

That is the capability Constraint Geometry is designed to evaluate.

9. Dual-Target Architecture

The central proposal of Constraint Geometry is that outcome recognition and mechanism attribution should be evaluated separately.

Traditional benchmark architectures typically evaluate a single target.

A model receives observations and is scored according to whether it predicts the correct outcome.

This approach is appropriate when outcome prediction is the capability of interest.

Constraint Geometry introduces a different possibility.

Rather than treating outcome prediction as the sole evaluation target, benchmark tasks can be designed to evaluate two related but distinct targets simultaneously:

Target A: Outcome Recognition

and

Target B: Mechanism Attribution

The purpose of this architecture is not to assume that these targets represent different capabilities.

Rather, it is to create a benchmark environment in which their relationship can be measured empirically.

If outcome prediction and mechanism attribution are effectively the same task, performance should remain closely coupled.

If meaningful separation exists, the benchmark should reveal that separation.

The dual-target architecture therefore transforms a conceptual question into a measurable one.

9.1 Why Two Targets?

The argument for dual-target evaluation follows directly from the previous sections.

Outcome recognition and mechanism attribution answer different questions.

Outcome recognition asks:

What happened?

Mechanism attribution asks:

Why did it happen?

In many benchmark settings these questions are treated as though they are interchangeable.

Constraint Geometry treats them as distinct evaluation targets.

The objective is not to assume separation.

The objective is to measure it.

Dual-target evaluation therefore provides a framework for investigating whether successful outcome recognition necessarily implies successful mechanism attribution, or whether meaningful divergence can occur between them.

9.2 Target A: Outcome Recognition

Target A evaluates conventional benchmark performance.

The objective is to determine whether a model can correctly identify the observable outcome associated with a trajectory.

Examples include:

- Stable vs. unstable
- Recovery vs. failure
- Progression vs. stabilization
- High risk vs. low risk

Target A therefore evaluates state recognition.

The benchmark question is:

What happened?

This target aligns closely with traditional benchmark design and provides a familiar reference point against which mechanism-attribution performance can be compared.

Constraint Geometry does not replace outcome prediction.

Outcome recognition remains a valuable benchmark target.

The framework simply evaluates an additional target alongside it.

9.3 Target B: Mechanism Attribution

Target B evaluates mechanism attribution.

The objective is to determine which structural explanation is most consistent with an observed trajectory.

Examples include:

- Repair Capacity Collapse
- Signal Readability Collapse
- Alignment Failure
- Compensation Exhaustion
- Dominance Shift
- Latency Failure

Target B therefore evaluates structural explanation rather than outcome recognition.

The benchmark question is:

Why did it happen?

More precisely:

Which mechanism best explains the observed trajectory?

Unlike outcome labels, mechanism labels refer to the structural conditions responsible for generating behavior.

The benchmark therefore evaluates whether a model can distinguish among competing explanations that remain consistent with similar observations.

9.4 Outcome–Mechanism Decoupling

The central measurement objective of the dual-target architecture is outcome–mechanism decoupling.

Outcome–mechanism decoupling refers to situations in which performance on outcome recognition diverges from performance on mechanism attribution.

Such divergence becomes possible whenever:

- Different mechanisms produce similar outcomes
- Similar mechanisms produce different outcomes
- Compensation obscures structural conditions
- Dominance relationships remain ambiguous
- Partial observability preserves uncertainty

These conditions create situations in which identifying an outcome does not automatically identify its explanation.

The extent of this divergence becomes an empirical result rather than a theoretical assumption.

If models consistently perform well on both targets, outcome recognition and mechanism attribution may be closely linked.

If performance diverges, the benchmark reveals that outcome prediction and mechanism attribution capture different aspects of reasoning.

For this reason, outcome–mechanism decoupling represents one of the primary quantities of interest within the framework.

9.5 Interpreting Dual-Target Performance

Dual-target evaluation creates a richer interpretation space than conventional benchmark scoring.

Consider four possible outcomes:

Outcome Recognition	Mechanism Attribution	Interpretation
High	High	Outcome and mechanism both identified correctly
High	Low	Correct prediction, incorrect explanation
Low	High	Correct mechanism, uncertain outcome
Low	Low	Neither target identified correctly

Traditional benchmarks distinguish primarily between correct and incorrect outcomes.

Dual-target evaluation introduces an additional dimension.

It becomes possible to evaluate not only whether a model predicts correctly, but whether it predicts for structurally appropriate reasons.

This distinction is particularly important when benchmark performance is interpreted as evidence of reasoning, explanation, or understanding.

9.6 Measuring the Two Targets

Constraint Geometry evaluates two forms of performance.

Outcome Accuracy

Outcome Accuracy measures performance on Target A.

This metric evaluates whether the observable outcome associated with a trajectory is correctly identified.

Examples include:

- Classification accuracy
- Precision
- Recall
- F1 score

These metrics remain unchanged from conventional benchmark evaluation.

Structural Alignment Accuracy (SAA)

Structural Alignment Accuracy measures performance on Target B.

SAA evaluates the degree to which a model's inferred structural explanation corresponds to the generative mechanism responsible for a trajectory.

Conceptually, SAA plays a role analogous to outcome accuracy.

The difference is that the target is structural explanation rather than outcome.

A prediction may therefore be:

- Outcome-correct but mechanism-incorrect
- Mechanism-correct but outcome-incorrect
- Correct on both targets
- Incorrect on both targets

This distinction allows structural reasoning performance to be measured independently of outcome recognition.

9.7 Measuring Outcome–Mechanism Decoupling

Outcome–Mechanism Decoupling can itself be measured.

Conceptually, decoupling captures the extent to which outcome recognition and mechanism attribution are solved independently.

When Outcome Accuracy and Structural Alignment Accuracy remain closely aligned, the two targets may be strongly coupled.

When substantial divergence emerges, the benchmark provides evidence that outcome recognition and mechanism attribution are not being solved identically.

A model may correctly identify outcomes while failing to identify the mechanisms responsible for them.

Conversely, a model may correctly identify governing mechanisms even when outcome prediction remains uncertain.

The magnitude and frequency of such divergences provide an empirical measure of outcome–mechanism decoupling.

For this reason, the benchmark is designed not merely to evaluate performance on two targets, but to evaluate the relationship between them.

9.8 Structural Alignment

Structural Alignment refers to the degree of correspondence between an inferred explanation and the mechanism responsible for generating a trajectory.

Within the benchmark framework, alignment is assessed against the known generative structure used to construct benchmark cases.

This is one of the advantages of synthetic benchmark environments.

The benchmark designer knows which mechanism generated each trajectory.

As a result, mechanism attribution can be evaluated directly rather than inferred indirectly.

Structural Alignment therefore transforms explanation quality into a measurable benchmark variable.

It provides the conceptual foundation from which Structural Alignment Accuracy (SAA) is derived.

9.9 Why Synthetic Benchmarks Matter

Mechanism attribution is difficult to evaluate in many real-world systems because the true governing mechanism is often unknown.

Outcomes may be observable while the structural conditions responsible for those outcomes remain uncertain, contested, or only partially understood.

Synthetic benchmark environments provide an important advantage.

Within a synthetic benchmark, the generative mechanism is specified during benchmark construction.

The benchmark designer therefore knows which constraints, relationships, compensatory processes, dominance structures, and trajectory dynamics produced each case.

This allows mechanism attribution to be evaluated directly rather than inferred indirectly from outcomes.

The value of synthetic environments is not that they perfectly represent real-world systems.

Rather, it is that they provide controlled conditions in which structural explanations are known and benchmark evaluation becomes unambiguous.

Constraint Geometry relies on this property to transform mechanism attribution from a conceptual objective into a measurable benchmark target.

9.10 Benchmark Implications

The introduction of dual-target evaluation changes the interpretation of benchmark performance.

Traditional evaluation asks:

Can the model identify the outcome?

Constraint Geometry asks:

Can the model identify both the outcome and the mechanism responsible for it?

This distinction expands the evaluation space.

Benchmark performance is no longer summarized solely by outcome accuracy.

It becomes possible to evaluate:

- Outcome recognition
- Mechanism attribution
- Outcome–mechanism decoupling
- Structural alignment
- Failure reconstruction
- Governing constraint identification

The result is a benchmark architecture designed not only to measure what happened, but also to evaluate whether a model can identify why it happened.

More importantly, it becomes possible to evaluate not only whether a model predicts correctly, but whether it predicts for structurally appropriate reasons.

The dual-target framework therefore serves as the evaluation foundation of the Constraint Geometry Benchmark Suite and provides the bridge between benchmark construction and empirical investigation.

Most importantly, it converts the central hypothesis of this paper into a measurable question:

Is mechanism attribution simply another form of outcome recognition, or does it represent a distinct benchmark target?

The purpose of the benchmark is not to assume the answer.

It is to measure it.

Part V — Demonstration Domain

10. Why Oncology?

The preceding sections introduced Constraint Geometry as a benchmark framework for evaluating mechanism attribution under conditions of structural ambiguity. A benchmark framework, however, requires more than evaluation metrics. It also requires an environment in which the target capability can be expressed, measured, and controlled.

Constraint Geometry evaluates whether models can distinguish among competing structural explanations that remain consistent with similar observations. The challenge is therefore to identify a domain that naturally supports interacting constraints, compensation, partial observability, delayed effects, competing mechanisms, and intervention-relevant structure.

This paper uses synthetic oncology-inspired environments for that purpose.

The choice is methodological rather than clinical.

The framework is not intended to establish biological truths, model disease progression, evaluate treatments, or assess medical competence. Instead, oncology serves as a demonstration environment within which mechanism-attribution tasks can be constructed under controlled conditions.

The significance of oncology within this framework is therefore not biomedical.

It is architectural.

The domain provides a rich benchmark ecology in which structural ambiguity can be preserved while remaining interpretable and measurable.

10.1 Benchmark Ecology

Different domains support different forms of evaluation.

Some environments naturally emphasize prediction.

Others emphasize planning, optimization, adaptation, or control.

Constraint Geometry requires a domain that supports mechanism attribution.

More specifically, it requires an environment characterized by:

- Multiple interacting constraints
- Compensation
- Partial observability
- Delayed effects
- Competing explanations
- Intervention relevance
- Structural ambiguity

Together, these properties create situations in which multiple explanations remain plausible despite access to the same observations.

This property is essential because mechanism attribution becomes meaningful only when alternative explanations remain viable.

Within this paper, **benchmark ecology** refers to the structural properties of an environment that make a particular form of reasoning measurable.

Oncology provides a benchmark ecology that naturally supports mechanism-attribution evaluation.

10.2 Multiple Interacting Systems

Many benchmark environments are dominated by relatively small numbers of variables or clearly defined relationships.

Oncology differs.

Behavior emerges through interactions among multiple partially coupled processes operating across different temporal and spatial scales.

For benchmark construction, five broad process categories are particularly useful:

Immune Processes

Immune activity influences recognition, suppression, adaptation, and response.

Repair Processes

Repair mechanisms maintain structural integrity and recover function following stress or damage.

Inflammatory Processes

Inflammatory signaling influences communication, adaptation, and resource allocation throughout the system.

Metabolic Processes

Metabolic conditions shape resource availability, energetic capacity, and resilience.

Signaling Processes

Signaling systems coordinate information flow across the broader system.

Signal degradation, latency, distortion, and unreadability may alter behavior even when other capacities remain intact.

The importance of these categories lies not in their biological details but in the fact that they create multiple interacting sources of constraint.

The benchmark task therefore becomes one of identifying which constraints govern an observed trajectory.

10.3 Rich Constraint Interaction

Constraint Geometry evaluates relationships among constraints rather than isolated variables.

Oncology provides an unusually rich environment for expressing such relationships.

Immune activity may compensate for declining repair capacity.

Metabolic constraints may alter signaling effectiveness.

Inflammatory processes may amplify or suppress adaptive responses.

Information may remain available while becoming increasingly difficult to interpret.

Observable behavior therefore emerges from interactions among processes rather than from any single process considered independently.

This characteristic naturally supports relationship-centric reasoning.

The benchmark question becomes:

Which configuration of interacting constraints best explains the observed trajectory?

rather than:

Which variable changed?

10.4 Structural Ambiguity by Construction

The value of oncology is not complexity itself.

The value is that complexity generates structural ambiguity.

Multiple active constraints, delayed effects, compensation, and partial observability create situations in which several explanations remain consistent with the same observations.

These conditions directly support the central design objective of Constraint Geometry.

Mechanism attribution becomes meaningful only when competing explanations remain plausible.

If observations uniquely identify the correct explanation, attribution becomes trivial.

If outcomes uniquely determine mechanisms, attribution collapses into prediction.

Oncology-inspired environments naturally support situations in which:

- Different mechanisms produce similar outcomes
- Similar mechanisms produce different outcomes
- Compensation conceals instability
- Dominance relationships shift through time
- Multiple explanations remain viable

These properties preserve structural ambiguity by construction.

10.5 Compensation and Hidden Instability

Compensation plays a central role throughout the Constraint Geometry framework.

The benchmark architecture relies on situations in which observable outcomes remain partially decoupled from underlying structural conditions.

Oncology-inspired environments readily support such scenarios.

Systems may remain stable despite accumulating structural stress.

Alternative processes may temporarily offset declining capacity elsewhere.

Redundant pathways may conceal emerging vulnerabilities.

Observable deterioration may be delayed long after underlying instability has begun.

These properties create conditions in which apparent stability and actual stability diverge.

Such divergence is valuable because it preserves uncertainty regarding the correct structural explanation.

Without compensation, outcomes often become direct reflections of underlying conditions.

With compensation, mechanism attribution remains meaningful.

10.6 Why Not Simpler Domains?

A natural question is why a benchmark framework designed to evaluate mechanism attribution requires a relatively rich domain at all.

Why not use simpler systems?

The answer is that simple environments often fail to preserve structural ambiguity.

When systems contain only a small number of variables or relationships, mechanisms frequently become recoverable directly from outcomes.

In such cases, mechanism attribution collapses into prediction.

The objective is not to maximize complexity.

Complexity alone does not improve benchmark quality.

The objective is to create situations in which multiple structural explanations remain plausible while still allowing correct attribution.

Richer environments make this easier to achieve.

They provide the structural diversity necessary for meaningful mechanism-attribution tasks without requiring unrestricted real-world complexity.

10.7 Why Oncology Is Useful for Benchmark Construction

The value of oncology as a benchmark domain derives from its structural properties rather than its clinical importance.

From a benchmark-design perspective, oncology naturally satisfies several requirements introduced earlier in this paper.

Benchmark Requirement	Oncology Provides
Many-to-One Mapping	Multiple mechanisms producing similar outcomes
One-to-Many Mapping	Similar mechanisms producing different trajectories
Structural Ambiguity	Competing explanations consistent with observations
Compensation	Stability despite hidden instability
Partial Observability	Incomplete access to governing processes
Dominance Competition	Multiple active constraints influencing behavior
Intervention Relevance	Different mechanisms imply different actions

These properties align closely with the design principles underlying Constraint Geometry.

The domain therefore provides a useful environment for evaluating whether models can distinguish among competing structural explanations.

10.8 Benchmark Construction Rather Than Biological Modeling

The benchmark framework uses oncology-inspired environments because they provide convenient structural ingredients for benchmark design.

The benchmark suite does not attempt to model oncology comprehensively.

Nor does it attempt to reproduce the full complexity of biological systems.

The environment is intentionally simplified.

Its purpose is to create controlled situations in which mechanism attribution can be measured directly.

The framework therefore prioritizes interpretability and evaluability over biological realism.

This distinction is important because the contribution of this work concerns benchmark architecture rather than biomedical modeling.

10.9 Why Synthetic Oncology Rather Than Real Oncology?

A natural question is why the benchmark framework uses synthetic oncology-inspired environments rather than real-world oncology datasets.

The answer follows directly from the evaluation objectives introduced in earlier sections.

Mechanism attribution requires ground-truth mechanisms.

In many real-world systems, including oncology, the true governing mechanism responsible for an observed outcome may be uncertain, partially observed, contested, or unknown.

Outcomes are often observable.

Mechanisms are not.

Real-world datasets therefore present a fundamental challenge for mechanism-attribution evaluation: benchmark designers may not know the correct structural explanation against which model performance should be assessed.

Synthetic benchmark environments provide an important advantage.

Within a synthetic environment, the generating mechanisms are specified during benchmark construction.

The benchmark designer therefore knows which constraints, compensatory processes, dominance relationships, signaling dynamics, and structural conditions produced each trajectory.

This allows mechanism attribution to be evaluated directly rather than inferred indirectly.

The value of synthetic oncology-inspired environments is therefore not that they perfectly represent biological reality.

It is that they provide controlled conditions in which structural explanations are known, benchmark targets are unambiguous, and mechanism attribution becomes measurable.

This property is essential for Structural Alignment Accuracy, Outcome–Mechanism Decoupling, and the broader dual-target evaluation architecture introduced earlier in this paper.

10.10 What This Paper Does Not Claim

The use of oncology-inspired environments may invite interpretations that extend beyond the scope of the framework.

Several clarifications are therefore important.

This paper does not propose:

A Cancer Model

The benchmark suite is not intended to represent the true mechanisms of cancer.

A Diagnostic Framework

The benchmark does not diagnose disease or classify patients.

A Treatment Framework

The benchmark does not recommend therapies or interventions for clinical use.

A Biological Theory

The framework does not claim to explain biological behavior or establish causal relationships within living systems.

A Clinical Decision System

The benchmark is not designed for patient management or clinical deployment.

Evidence of Medical Efficacy

Performance within synthetic benchmark environments should not be interpreted as evidence of real-world clinical competence.

The contribution of this paper is substantially narrower.

It introduces a benchmark-generation methodology designed to evaluate mechanism attribution under controlled conditions.

10.11 Domain Choice as a Benchmark Design Decision

The selection of oncology should ultimately be understood as a benchmark-design decision rather than a scientific claim.

The framework requires a benchmark ecology characterized by:

- Multiple interacting constraints
- Compensation
- Partial observability
- Structural ambiguity
- Mechanism diversity
- Intervention relevance

Oncology provides these characteristics in a form that is intuitive, familiar, and highly amenable to benchmark construction.

The significance of the domain therefore lies not in its biomedical content but in its ability to support controlled evaluation of structural reasoning.

The benchmark question remains unchanged:

Can a model distinguish among competing structural explanations that produce similar observable outcomes?

Oncology is simply one environment in which that question can be asked.

The framework itself is intended to generalize beyond oncology and apply wherever mechanism attribution matters.

The next section introduces the benchmark family itself and describes how these principles are translated into concrete benchmark tasks.

11. The Constraint Geometry Benchmark Family

The preceding sections introduced the conceptual foundations of Constraint Geometry, established the rationale for mechanism-attribution evaluation, defined the dual-target architecture, and described the design principles required to preserve structural ambiguity.

This section introduces the Constraint Geometry Benchmark Suite (CGBS), a family of synthetic benchmark environments designed to evaluate progressively richer forms of structural reasoning.

The benchmark family should be understood as a sequence of structural layers rather than a collection of independent datasets.

Each benchmark isolates a distinct source of structural ambiguity while preserving the broader design principles of the framework.

Earlier benchmarks focus on basic constraint interaction.

Later benchmarks introduce compensation, information degradation, latency, partial observability, dominance competition, and mechanism ambiguity.

Together these layers progressively increase the difficulty of mechanism attribution while maintaining the central objective of the framework:

to evaluate whether models can distinguish among competing structural explanations that remain consistent with similar observations.

The benchmarks are not intended to represent stages of disease progression or increasing biological realism.

Instead, they represent progressively richer mechanism-attribution challenges.

11.0 Benchmark Layer Logic

The benchmark family follows a cumulative design philosophy.

Each benchmark introduces a new source of structural ambiguity while retaining those introduced previously.

As a result, the benchmark progression does not merely increase difficulty.

It increases the number of structural explanations that remain plausible for a given observation.

The progression therefore reflects increasing mechanism-attribution complexity rather than increasing predictive difficulty.

Viewed collectively, the benchmark suite functions as a layered mechanism-attribution curriculum.

Each benchmark introduces a new structural question, a new attribution challenge, and a new mechanism through which outcome recognition and mechanism attribution may diverge.

11.1 v0.1 Constraint Geometry

Benchmark Question

Is instability present within the system?

Target Structure

Basic constraint interaction and stability classification.

Mechanism Targets

- Stable constraint configuration
- Unstable constraint configuration

Representative Case Types

- Stable systems
- Unstable systems
- High-load stable systems
- Low-load unstable systems

Anti-Shortcut Design

- No single-variable separation
- High burden does not always imply instability
- Low burden does not always imply stability

Scoring Architecture

Target A

Outcome Accuracy

- Stable
- Unstable

Target B

Structural Alignment Accuracy

- Correct identification of governing stability structure

Source of Structural Ambiguity

Constraint interaction.

Observable burden and structural stability are intentionally decoupled.

Structural Contribution

v0.1 establishes the foundational distinction between observable burden and structural stability.

The benchmark evaluates whether models can identify instability as a property of constraint relationships rather than individual variables.

11.2 v0.2 Compensation Geometry

Benchmark Question

Is compensation masking instability?

Target Structure

Compensatory processes operating between constraints and outcomes.

Mechanism Targets

- Genuine stability
- Compensated stability
- Compensation exhaustion
- Compensation failure

Representative Case Types

- Stable systems
- Stable compensated systems
- Unstable uncompensated systems
- Compensation-collapse systems

Anti-Shortcut Design

- Apparently stable systems may be unstable
- Observable outcomes do not uniquely determine mechanisms
- Compensation occasionally overrides dominant signals

Scoring Architecture

Target A

- Stable
- Unstable

Target B

- Compensation state attribution

Structural Alignment Accuracy

Correct identification of the compensatory mechanism governing the trajectory.

Source of Structural Ambiguity

Compensation.

Observable stability and underlying stability may diverge.

Structural Contribution

v0.2 introduces the first major source of hidden instability.

The benchmark evaluates whether models can distinguish observable behavior from underlying structural conditions.

11.3 v0.3 Readability Collapse

Benchmark Question

Can the system still interpret critical signals?

Target Structure

Signal interpretation and information-processing integrity.

Mechanism Targets

- Readable signal environment
- Partial readability degradation
- Severe readability collapse
- Noise-dominated signaling

Representative Case Types

- High-signal systems
- Signal-noise conflict systems
- Progressive readability degradation
- Misinterpretation-dominated systems

Anti-Shortcut Design

- Signal presence does not imply signal usability
- Missing signals and unreadable signals may appear similar
- Similar outcomes emerge from different information conditions

Scoring Architecture

Target A

- Functional
- Dysfunctional

Target B

- Readability state attribution

Structural Alignment Accuracy

Correct identification of the signal-processing condition responsible for the trajectory.

Source of Structural Ambiguity

Readability.

Signal availability and signal interpretability become separable.

Structural Contribution

v0.3 introduces information quality as an independent source of instability.

The benchmark evaluates whether models distinguish signal availability from signal interpretability.

11.4 v0.4 Signal Alignment

Benchmark Question

Are repair signals aligned with system needs?

Target Structure

Alignment between system responses and governing constraints.

Mechanism Targets

- High alignment
- Partial alignment
- Misalignment
- Severe misdirection

Representative Case Types

- Correctly aligned repair
- Resource misallocation
- Secondary-target fixation
- Dominant-constraint neglect

Anti-Shortcut Design

- Strong responses may still be ineffective
- Resource abundance does not imply alignment
- Similar outcomes may emerge from different alignment states

Scoring Architecture

Target A

- Recovery
- Failure

Target B

- Alignment attribution

Structural Alignment Accuracy

Correct identification of the relationship between system action and system need.

Source of Structural Ambiguity

Alignment.

Activity and effectiveness become distinguishable.

Structural Contribution

v0.4 introduces the distinction between acting and acting on the correct constraint.

The benchmark evaluates whether models can identify misdirected adaptation.

11.5 v0.5 Signal Latency Boundary

Benchmark Question

Has delay exceeded recoverable limits?

Target Structure

Timing-dependent stability and delayed intervention effects.

Mechanism Targets

- Timely response
- Delayed response
- Recoverable delay
- Irrecoverable delay

Representative Case Types

- Early intervention
- Borderline recovery windows
- Late intervention
- Missed recovery windows

Anti-Shortcut Design

- Identical signals produce different outcomes depending on timing
- Similar outcomes may arise from different delay structures
- Latency remains partially independent of burden

Scoring Architecture

Target A

- Recovery
- Failure

Target B

- Latency attribution

Structural Alignment Accuracy

Correct identification of timing as the governing mechanism.

Source of Structural Ambiguity

Latency.

Identical structural states may produce different outcomes depending on timing.

Structural Contribution

v0.5 introduces temporal structure into mechanism attribution.

The benchmark evaluates whether models recognize timing as a governing constraint.

11.6 v0.6 Missing Signal Detection

Benchmark Question

Is a critical signal absent?

Target Structure

Reasoning under incomplete information.

Mechanism Targets

- Complete observability
- Partial observability
- Critical signal absence
- Observability collapse

Representative Case Types

- Fully observed systems
- Missing but irrelevant signals
- Missing critical signals
- Ambiguous observability states

Anti-Shortcut Design

- Missing information does not always imply instability
- Stable systems may operate under incomplete information
- Absence and unreadability are not identical conditions

Scoring Architecture

Target A

- Stable
- Unstable

Target B

- Observability attribution

Structural Alignment Accuracy

Correct identification of the role played by missing information.

Source of Structural Ambiguity

Partial observability.

Incomplete information becomes separable from instability.

Structural Contribution

v0.6 introduces uncertainty arising from incomplete observation.

The benchmark evaluates whether models distinguish ignorance from instability.

11.7 v0.7 Constraint Dominance

Benchmark Question

Which constraint governs the trajectory?

Target Structure

Dominance relationships among competing constraints.

Mechanism Targets

- Repair-dominant
- Signaling-dominant
- Metabolic-dominant
- Inflammatory-dominant
- Mixed-dominance states

Representative Case Types

- Clear dominance
- Contested dominance
- Dominance switching
- Multi-constraint competition

Anti-Shortcut Design

- Dominance remains ambiguous in some cases
- Multiple constraints remain simultaneously active
- Similar outcomes arise from different dominant mechanisms

Scoring Architecture

Target A

- Outcome classification

Target B

- Dominant mechanism attribution

Structural Alignment Accuracy

Correct identification of the governing constraint responsible for the trajectory.

Source of Structural Ambiguity

Dominance competition.

Multiple plausible governing mechanisms remain active simultaneously.

Structural Contribution

v0.7 represents the culmination of the benchmark family.

The benchmark evaluates whether models can identify the dominant mechanism governing system behavior under conditions of structural ambiguity.

Benchmark Progression

The benchmark family forms a progression from basic structural classification toward explicit mechanism attribution.

Benchmark	Primary Question	Source of Structural Ambiguity	Mechanism Target
v0.1 Constraint Geometry	Is instability present?	Constraint interaction	Stability structure
v0.2 Compensation Geometry	Is compensation masking instability?	Compensation	Compensation attrib
v0.3 Readability Collapse	Can the system still interpret signals?	Readability	Signal interpretation
v0.4 Signal Alignment	Are responses aligned with need?	Alignment	Response attributio

v0.5 Signal Latency Boundary	Has delay exceeded recoverable limits?	Latency	Timing attribution
v0.6 Missing Signal Detection	Is a critical signal absent?	Partial observability	Observability attribu
v0.7 Constraint Dominance	Which constraint governs the trajectory?	Dominance competition	Mechanism attributi

Together these benchmarks progressively increase structural ambiguity while preserving the central design objective of the framework:

to evaluate whether models can distinguish among competing structural explanations that remain consistent with similar observations.

The benchmark family should therefore be understood as a layered mechanism-attribution curriculum.

Each benchmark introduces a new source of structural ambiguity and a corresponding attribution task.

Together they create a controlled environment in which outcome recognition, mechanism attribution, structural alignment, and outcome–mechanism decoupling can be evaluated systematically.

The benchmark suite does not assume that mechanism attribution constitutes a distinct capability.

Rather, it provides a structured environment in which that possibility can be measured empirically.

In this sense, the Constraint Geometry Benchmark Suite operationalizes the central hypothesis of this paper:

that mechanism attribution may represent a measurable benchmark target distinct from outcome recognition alone.

Part VI — Constraint Dominance

12. Dominance as a Benchmark Task

The preceding sections argued that mechanism attribution becomes meaningful when multiple structural explanations remain plausible. The benchmark family introduced in Section 11 progressively increased structural ambiguity through compensation, readability degradation, latency, partial observability, and competing constraint interactions.

These benchmark layers culminate in a more specific attribution problem:

Which constraint governs the trajectory of the system?

This question introduces the concept of dominance.

Constraint Geometry treats dominance as a benchmark target rather than a theoretical assumption. The objective is not merely to identify which constraints are present within a system. The objective is to determine which constraint exerts the greatest influence over future system behavior.

This distinction is important because most complex systems contain multiple active constraints simultaneously.

Repair processes, signaling processes, metabolic conditions, inflammatory responses, resource limitations, information bottlenecks, and timing effects may all influence behavior at the same time.

Mechanism attribution therefore requires more than identifying constraints.

It requires determining which constraints matter most.

Constraint dominance provides a formal benchmark framework for evaluating that capability.

12.1 The Dominance Principle

Mechanism attribution requires more than identifying which constraints are present within a system.

It requires identifying which constraints most strongly shape the space of future trajectories.

Constraint Geometry refers to this property as **dominance**.

The central premise of dominance evaluation is that systems are often influenced by many constraints simultaneously, but governed by only a subset of them.

The objective of mechanism attribution is therefore not merely to identify active constraints.

It is to identify governing constraints.

Dominance provides the benchmark-level bridge between structural ambiguity and mechanism attribution.

Earlier benchmark layers introduced competing explanations.

Dominance asks which explanation ultimately matters most.

12.2 Why Dominance Matters

Many benchmark tasks evaluate the presence or absence of specific conditions.

Constraint Geometry evaluates something different.

The framework asks whether a model can distinguish between:

- Constraints that influence behavior
- Constraints that govern behavior

This distinction is not always obvious.

Multiple constraints may appear equally important when viewed through observable variables alone.

Some constraints may exert substantial local influence while having limited effect on overall trajectory.

Others may appear relatively minor while fundamentally determining the future state of the system.

Mechanism attribution therefore requires reasoning about relative influence rather than simple presence.

The benchmark question becomes:

Which constraint most strongly shapes the trajectory of the system?

This question forms the foundation of dominance evaluation.

12.3 The Dominant Constraint

A dominant constraint is the constraint whose modification would most substantially alter the future trajectory of the system relative to competing constraints.

Dominance is therefore defined by **counterfactual influence rather than observable magnitude**.

A dominant constraint is not necessarily:

- The largest signal
- The most abnormal variable
- The most visible failure
- The most active process

Instead, it is the constraint that most strongly governs the space of available trajectories.

For example:

A system may exhibit substantial inflammatory activity.

At the same time, a signaling failure may prevent adaptive responses from reaching the processes responsible for repair.

Although inflammation appears more visible, signaling may be the dominant constraint because correcting signaling would alter the trajectory more profoundly than reducing inflammation.

Constraint Geometry therefore separates:

Observed Importance

from

Structural Importance

Dominance concerns structural importance.

12.4 Secondary Constraints

Most systems contain multiple meaningful constraints.

Not all are dominant.

Secondary constraints are constraints that influence system behavior but do not primarily determine trajectory.

These constraints may:

- Amplify dominant effects
- Buffer dominant effects
- Modify local behavior
- Become dominant later in time

Secondary constraints remain important because they often contribute to structural ambiguity.

A benchmark containing only dominant constraints quickly becomes deterministic.

Constraint Geometry therefore incorporates situations in which multiple constraints remain active simultaneously.

Mechanism attribution requires distinguishing between dominant and secondary influences.

This distinction increases benchmark difficulty while preserving structural realism and diversity.

12.4.1 Dominance Is Relational

Dominance is not an intrinsic property of a constraint.

A constraint becomes dominant only relative to other active constraints within a particular structural configuration.

A signaling constraint may govern one trajectory while remaining secondary in another.

Likewise, a repair constraint may dominate during one phase of system behavior and become subordinate during another.

Dominance therefore emerges from relationships among constraints rather than from any individual constraint considered in isolation.

This property makes dominance a structural concept rather than a variable-level concept.

It also reflects a broader theme running throughout Constraint Geometry:

System behavior is governed not by isolated variables, but by patterns of interaction among constraints.

12.5 Dominance Margin

Dominance is rarely binary.

Some systems contain a clearly dominant constraint.

Others contain several constraints exerting comparable influence.

Constraint Geometry therefore introduces the concept of **dominance margin**.

Dominance margin refers to the degree of separation between the dominant constraint and competing constraints.

High Dominance Margin

One constraint clearly governs the trajectory.

Mechanism attribution is relatively straightforward.

Moderate Dominance Margin

Several constraints contribute significantly to system behavior.

Attribution becomes more difficult.

Low Dominance Margin

Multiple constraints remain plausible governing mechanisms.

Attribution becomes inherently ambiguous.

Dominance margin therefore provides a continuous measure of attribution difficulty.

As dominance margin decreases, competing explanations become increasingly plausible and mechanism attribution becomes progressively harder.

From a benchmark-design perspective, dominance margin functions as a controllable generator of structural ambiguity.

It provides a systematic way to vary attribution difficulty while preserving the underlying benchmark structure.

12.6 Compensation and Dominance

Compensation complicates dominance attribution.

A dominant constraint may be partially concealed by compensatory processes.

Observable behavior may therefore suggest one explanation while the governing constraint lies elsewhere.

For example:

A declining repair process may remain hidden because compensatory mechanisms temporarily preserve function.

Observed stability may suggest that repair remains sufficient.

The underlying trajectory may indicate otherwise.

Compensation therefore creates situations in which:

- Apparent dominance differs from actual dominance
- Visible constraints differ from governing constraints
- Outcome recognition diverges from mechanism attribution

This property makes compensation one of the most important sources of dominance ambiguity.

Within the benchmark framework, compensation prevents dominance attribution from collapsing into simple signal ranking.

12.7 Contested Dominance

Not all systems possess a single clearly dominant constraint.

In some cases, multiple constraints compete for influence.

Constraint Geometry refers to these situations as **contested dominance states**.

Examples include:

- Repair and signaling exerting comparable influence
- Metabolic and inflammatory constraints jointly shaping trajectory
- Compensation delaying emergence of a clearly dominant mechanism

In contested dominance states:

- Multiple explanations remain plausible
- Structural ambiguity increases
- Attribution becomes more difficult

Contested dominance represents one of the clearest manifestations of structural ambiguity within the framework.

Multiple explanations remain simultaneously plausible because influence is distributed across competing constraints rather than concentrated within a single governing mechanism.

These situations are particularly valuable for benchmark construction because they resist shortcut solutions.

A model cannot simply identify the largest signal.

Instead, it must reason about competing structural explanations and their relative influence over future trajectories.

Contested dominance therefore represents one of the highest-complexity mechanism-attribution tasks within the benchmark family.

12.8 Dominance Switching

Dominance may also change through time.

A secondary constraint may become dominant.

A dominant constraint may be mitigated.

Compensation may temporarily suppress one constraint while amplifying another.

The governing mechanism of a trajectory is therefore not always static.

Examples include:

- Signaling failure initially dominates but later gives way to repair collapse
- Compensation initially governs outcomes but eventually becomes exhausted
- Resource limitations become dominant only after adaptation capacity declines

These situations introduce temporal structure into mechanism attribution.

The benchmark question expands from:

Which constraint governs the trajectory?

to:

Which constraint governs the trajectory now?

Dominance switching therefore represents a dynamic extension of dominance attribution and provides a natural bridge to future benchmark families involving intervention sequencing, adaptive control, and closed-loop stabilization.

12.9 Dominance as Mechanism Attribution

Dominance attribution represents the benchmark-level operationalization of mechanism attribution.

Earlier benchmark layers introduced distinct sources of structural ambiguity:

- Constraint interaction
- Compensation
- Readability degradation
- Alignment failure
- Latency
- Partial observability

Dominance asks which of these structural factors ultimately governs the trajectory.

This shift is important.

The benchmark no longer focuses primarily on the existence of structural conditions.

It focuses on identifying the structural condition that matters most.

Mechanism attribution therefore becomes an exercise in influence ranking rather than state recognition.

The objective is not to identify all active processes.

The objective is to identify the governing process.

12.10 Benchmark Implications

Dominance transforms mechanism attribution into a measurable benchmark task.

Within the dual-target architecture:

Target A

Outcome Recognition

Examples:

- Stable
- Unstable
- Recovery
- Failure

Target B

Dominance Attribution

Examples:

- Repair Dominant
- Signaling Dominant
- Metabolic Dominant
- Inflammatory Dominant
- Mixed Dominance

This distinction allows benchmark evaluation to investigate whether models can correctly identify governing constraints independently of outcome recognition.

The benchmark may therefore reveal situations in which:

- Outcome prediction succeeds
- Dominance attribution fails

Such cases provide direct evidence of outcome–mechanism decoupling.

They represent precisely the type of divergence that motivates the Constraint Geometry framework.

12.11 Dominance as the Culmination of the Benchmark Family

The benchmark family introduced in Section 11 progressively expanded the sources of structural ambiguity available within the evaluation environment.

Constraint interaction introduced ambiguity.

Compensation amplified ambiguity.

Readability, alignment, latency, and partial observability introduced additional layers of uncertainty.

Dominance integrates these layers into a single attribution task.

To identify the governing constraint, a model may need to reason simultaneously about:

- Constraint interaction
- Compensation
- Readability
- Alignment
- Latency
- Observability

Dominance therefore functions as the culmination of the benchmark family.

It represents the point at which mechanism attribution becomes most explicit.

The benchmark question is no longer:

What happened?

Nor even:

Why did it happen?

It becomes:

Which structural mechanism most strongly governed what happened?

Dominance therefore serves as the point at which the conceptual framework, benchmark architecture, and evaluation methodology converge.

Constraint interaction, compensation, readability, alignment, latency, and observability all contribute to structural ambiguity.

Dominance asks which of those factors ultimately governs the trajectory.

In this sense, dominance is not simply another benchmark target.

It is the benchmark-level expression of the central question motivating the entire framework:

Which mechanism matters most?

13. Mechanism Attribution Benchmarks

The preceding sections established mechanism attribution as the central evaluation target of the Constraint Geometry framework and identified dominance attribution as one of its most demanding forms. They further argued that outcome recognition and mechanism attribution should be evaluated separately through a dual-target architecture capable of measuring outcome–mechanism decoupling.

This section formalizes **Mechanism Attribution Benchmarks** as a benchmark category.

The objective is not merely to construct benchmark tasks that contain mechanisms.

The objective is to construct benchmark tasks in which successful performance requires identifying mechanisms.

This distinction is important.

Many benchmark tasks contain underlying mechanisms without explicitly evaluating whether models identify them.

Constraint Geometry proposes that structural explanations can themselves become benchmark targets.

The resulting benchmark architecture extends evaluation beyond outcome prediction toward direct assessment of explanation quality.

13.1 Explanation as a Benchmark Target

Traditional benchmark architectures primarily evaluate predictions.

Mechanism Attribution Benchmarks evaluate explanations.

The distinction is subtle but important.

Predictions concern outcomes.

Explanations concern the structural conditions that generate those outcomes.

Traditional benchmark evaluation typically asks:

What happened?

Mechanism Attribution Benchmarks ask:

Why did it happen?

More specifically:

Which structural explanation best accounts for what happened?

Constraint Geometry therefore treats explanation not as a byproduct of prediction, but as an independent evaluation target.

The objective is not merely to determine whether a model predicts correctly.

The objective is to determine whether it predicts for structurally appropriate reasons.

This shift forms the foundation of the benchmark category introduced in this section.

13.2 Outcome Prediction

Outcome prediction represents the dominant paradigm in contemporary benchmark design.

A model receives observations and attempts to predict an outcome.

Examples include:

- Disease progression
- Treatment response
- Recovery
- Failure
- Risk category
- System stability

The benchmark objective is straightforward:

Predict the correct outcome.

Performance is typically evaluated using:

- Accuracy
- Precision
- Recall
- F1 Score
- ROC-AUC
- Calibration metrics

These benchmarks have proven highly effective for evaluating predictive performance.

However, outcome prediction alone does not necessarily reveal whether a model has identified the mechanism responsible for the outcome.

A model may predict correctly while relying on:

- Correlations
- Proxies
- Statistical regularities
- Shortcut features

without identifying the governing structure of the system.

This observation does not diminish the value of outcome prediction.

Rather, it motivates a complementary benchmark category focused on explanation rather than outcome alone.

13.3 Mechanism Attribution

Mechanism attribution evaluates a different target.

Instead of predicting an outcome, the model identifies the structural explanation most consistent with the observed evidence.

The benchmark question becomes:

Which mechanism generated this trajectory?

Examples include:

- Repair Capacity Collapse
- Signal Readability Collapse
- Compensation Exhaustion
- Alignment Failure
- Latency Failure
- Dominance Shift

Unlike outcome labels, mechanism labels describe the structural conditions responsible for producing behavior.

Mechanism attribution therefore evaluates explanation rather than prediction alone.

Importantly, the objective is not to recover the complete generative structure of a system.

The objective is to identify the governing mechanism within a predefined set of competing structural explanations.

Constraint Geometry therefore occupies a middle ground between:

- Outcome prediction
- Full causal discovery

The benchmark evaluates mechanism attribution rather than unrestricted structural discovery.

13.3.1 Attribution vs Discovery

Mechanism attribution should not be confused with causal discovery.

The two objectives are related but distinct.

Causal discovery attempts to recover unknown generative structure from observations.

Mechanism attribution assumes a predefined set of candidate explanations and evaluates whether a model can identify the explanation most consistent with available evidence.

The benchmark therefore evaluates explanation selection rather than unrestricted structural discovery.

This distinction is important because Constraint Geometry does not claim to solve causal discovery.

Its objective is substantially narrower.

The framework evaluates whether models can discriminate among competing explanations when the space of possible mechanisms is known by construction.

13.4 Outcome Prediction vs Mechanism Attribution

Although related, outcome prediction and mechanism attribution evaluate different informational targets.

Outcome prediction asks:

What happened?

Mechanism attribution asks:

Why did it happen?

This distinction can be illustrated through a simple example.

Case	Outcome	Mechanism
A	Failure	Repair Capacity Collapse
B	Failure	Signal Readability Collapse

C	Failure	Compensation Exhaustion
D	Failure	Alignment Failure

From the perspective of outcome prediction, all four cases are identical.

From the perspective of mechanism attribution, they are distinct.

A benchmark focused solely on outcomes treats these cases as equivalent.

A mechanism-attribution benchmark does not.

This difference represents the central motivation for the framework.

Outcomes compress information.

Mechanism attribution attempts to recover information preserved within structural explanations.

The benchmark therefore evaluates distinctions that outcome labels alone cannot capture.

13.5 Dual-Target Evaluation

Constraint Geometry evaluates both outcome prediction and mechanism attribution simultaneously.

This dual-target architecture enables direct comparison between the two forms of performance.

Target A

Outcome Recognition

Examples:

- Stable
- Unstable
- Recovery
- Failure

Target B

Mechanism Attribution

Examples:

- Repair Collapse
- Readability Collapse
- Compensation Failure
- Dominance Shift

The benchmark therefore evaluates two related but distinct questions:

What happened?

and

Why did it happen?

The purpose of this architecture is not to assume that these tasks are different.

The purpose is to measure whether they are different.

If performance on both targets remains tightly coupled, this suggests limited separation.

If performance diverges, the benchmark reveals outcome–mechanism decoupling.

The degree of separation becomes an empirical result rather than a theoretical assumption.

13.6 Structural Alignment Accuracy

Mechanism attribution requires its own evaluation metric.

Traditional benchmark metrics evaluate outcome correctness.

Constraint Geometry introduces **Structural Alignment Accuracy (SAA)**.

SAA measures the degree to which a model's inferred explanation corresponds to the benchmark-generating mechanism responsible for a trajectory.

Conceptually, SAA plays the same role for explanations that classification accuracy plays for outcomes.

A trajectory is generated by a known mechanism.

The model produces an inferred explanation.

SAA evaluates the degree of correspondence between the two.

Conceptually:

```
Observed Trajectory
        ↓
Model Attribution
        ↓
Generating Mechanism
```

Correct alignment occurs when the inferred explanation matches the benchmark-generating mechanism.

Misalignment occurs when the model attributes the trajectory to the wrong structural explanation.

SAA therefore evaluates explanation quality directly rather than outcome correctness indirectly.

13.7 Outcome–Mechanism Divergence

One of the primary objectives of the benchmark family is to identify situations in which outcome prediction and mechanism attribution diverge.

Outcome–Mechanism Divergence represents the observable manifestation of outcome–mechanism decoupling within benchmark evaluation.

Several configurations are possible:

Outcome Prediction	Mechanism Attribution	Interpretation
Correct	Correct	Outcome and mechanism identified

Correct	Incorrect	Predictive success without explanation
Incorrect	Correct	Structural understanding despite outcome uncertainty
Incorrect	Incorrect	Neither target identified

The second row is particularly important.

A model may correctly predict an outcome while misidentifying the mechanism responsible for it.

Such cases demonstrate that outcome recognition alone cannot be assumed to imply mechanism attribution.

Large divergence between Outcome Accuracy and Structural Alignment Accuracy provides evidence that the two benchmark targets are not being solved identically.

The benchmark is therefore designed not only to measure performance on two targets, but to measure the relationship between them.

13.8 Attribution Failure Modes

Mechanism attribution introduces a new class of benchmark errors.

These errors arise not because the outcome is misidentified, but because the explanation is misidentified.

False Attribution

The model identifies a mechanism that did not generate the trajectory.

The outcome may still be predicted correctly.

Proxy Attribution

The model relies on shortcut features associated with a mechanism rather than identifying the mechanism itself.

This represents the mechanism-level analogue of proxy prediction.

Dominance Misidentification

The model identifies an active constraint but fails to identify the governing constraint.

The explanation is partially correct but structurally incomplete.

Compensation Blindness

The model fails to recognize compensatory processes that conceal underlying instability.

Observable behavior is interpreted literally rather than structurally.

Readability Confusion

The model confuses absent information with degraded information.

Signal absence and signal unreadability become conflated.

Equifinality Collapse

The model treats distinct mechanisms as equivalent because they produce similar outcomes.

Mechanistic diversity collapses into outcome categories.

These failure modes provide insight into how mechanism attribution breaks down and help distinguish structural reasoning failures from conventional prediction errors.

13.9 Mechanism Attribution as a Benchmark Category

Mechanism Attribution Benchmarks represent a distinct benchmark category because their primary target is structural explanation rather than outcome prediction.

They evaluate explanation quality directly rather than treating explanation as an implicit byproduct of predictive performance.

This distinction places the benchmark category between:

- Outcome prediction
- Causal discovery

The benchmark does not require complete recovery of generative structure.

Nor does it restrict evaluation to observable outcomes.

Instead, it evaluates whether models can discriminate among competing structural explanations under conditions of structural ambiguity.

The objective is not to determine whether a model can discover every cause.

The objective is to determine whether it can identify the governing explanation among plausible alternatives.

13.10 Benchmark Implications

Mechanism Attribution Benchmarks expand the scope of benchmark evaluation.

Traditional benchmarks primarily measure:

- State recognition
- Classification
- Forecasting
- Prediction

Mechanism Attribution Benchmarks additionally evaluate:

- Structural explanation
- Governing constraint identification
- Dominance attribution
- Failure reconstruction
- Intervention-relevant reasoning

This expansion does not replace outcome prediction.

Rather, it introduces an additional layer of evaluation.

The result is a benchmark architecture capable of measuring not only whether a model predicts correctly, but whether it identifies the structural explanation responsible for its prediction.

Traditional benchmarks ask:

What happened?

Mechanism Attribution Benchmarks ask:

Why did it happen?

More specifically:

Which structural explanation best accounts for what happened?

Constraint Geometry does not assume that answering this question requires a distinct capability.

It provides a benchmark architecture in which that possibility can be measured empirically.

In this sense, Mechanism Attribution Benchmarks represent the transition from evaluating outcomes to evaluating explanations, and from measuring prediction alone to measuring structural understanding.

14. Dominance Failure Modes

The previous sections introduced dominance attribution as the most explicit form of mechanism attribution within the Constraint Geometry framework. The central benchmark question was defined as:

Which constraint governs the trajectory of the system?

At first glance, this question appears straightforward.

In practice, however, dominance attribution is often difficult precisely because dominance itself may be ambiguous.

Real systems rarely present their governing mechanisms in a clear and unambiguous form. Multiple constraints may remain active simultaneously. Compensation may conceal the true source of instability. Dominance relationships may evolve through time. Apparent constraints may differ from governing constraints. Critical mechanisms may remain only partially observable.

As a result, identifying the governing constraint is often more difficult than identifying active constraints.

These difficulties are not incidental.

They define the benchmark problem.

Understanding dominance failure modes is important for two reasons.

First, they provide insight into why mechanism attribution is difficult.

Second, they provide a principled foundation for benchmark construction. Many of the most informative benchmark cases arise not when dominance is obvious, but when dominance attribution becomes uncertain, contested, or misleading.

Constraint Geometry therefore treats dominance failure modes not merely as errors, but as structured sources of ambiguity that make mechanism attribution measurable.

14.1 Why Dominance Attribution Fails

Dominance attribution fails whenever the relationship between observable signals and governing influence becomes uncertain.

The central challenge is that influence and appearance do not always coincide.

The most visible mechanism is not necessarily the most important mechanism.

The most active constraint is not necessarily the governing constraint.

The most abnormal signal is not necessarily the dominant source of system behavior.

Constraint Geometry identifies four broad sources of dominance ambiguity:

Visibility Ambiguity

The governing mechanism is partially hidden or observationally misleading.

Competitive Ambiguity

Multiple constraints exert comparable influence over system behavior.

Temporal Ambiguity

Dominance relationships evolve through time.

Compensation Ambiguity

Compensatory processes conceal the true source of instability.

The failure modes introduced in this section can be understood as specific manifestations of these broader categories.

Together they form a practical taxonomy of dominance ambiguity within the benchmark framework.

Visibility Ambiguity

14.2 False Dominance

False dominance occurs when **observational salience is mistaken for structural influence**.

A constraint appears to govern the trajectory but does not actually do so.

The constraint is visible.

The constraint is active.

The constraint may even be highly abnormal.

Yet it is not the primary driver of system behavior.

This creates a distinction between:

Apparent Dominance

and

Actual Dominance

For example:

A system may exhibit pronounced inflammatory activity.

Inflammation appears to be the dominant problem.

However, the true governing constraint may be a signaling failure that prevents adaptive responses from coordinating effectively.

Inflammation is visible.

Signaling failure is governing.

A model that identifies the most visible constraint rather than the governing constraint commits a false-dominance error.

False dominance represents the benchmark analogue of confounding at the mechanism-attribution level.

The observable signal attracts attention while obscuring the mechanism that actually governs the trajectory.

Constraint Geometry therefore includes benchmark cases in which apparent dominance and actual dominance deliberately diverge.

14.3 Hidden Dominance

Hidden dominance occurs when the governing constraint is not directly observable.

The dominant mechanism exists.

Its influence shapes the trajectory.

Yet the available observations provide only indirect evidence of its presence.

Examples include:

- Missing signals
- Delayed effects
- Latent structural damage
- Unobserved coordination failures

The system behaves as though a dominant constraint exists.

The constraint itself remains partially hidden.

This creates one of the most challenging attribution problems in the benchmark family.

The model must infer the governing mechanism from its effects rather than from direct observation.

Hidden dominance therefore combines:

- Mechanism attribution
- Partial observability
- Structural inference

within a single benchmark task.

Hidden dominance represents perhaps the purest mechanism-attribution challenge within the framework because the governing mechanism must be inferred from indirect effects rather than direct observation.

These cases play an important role because many real-world systems operate under incomplete information.

Competitive Ambiguity

14.4 Balanced Constraints

Balanced constraints occur when multiple constraints exert similar influence over system behavior.

No single constraint clearly governs the trajectory.

Several mechanisms remain plausible explanations.

For example:

- Repair capacity is declining
- Signal readability is deteriorating
- Metabolic stress is increasing

Each contributes meaningfully to system behavior.

None clearly dominates.

Balanced constraint states are important because they represent situations in which mechanism attribution becomes inherently uncertain.

Importantly, balanced constraints do not necessarily imply attribution failure.

Instead, they represent situations in which the correct structural interpretation may be that no single dominant constraint exists.

The benchmark objective is therefore not always to eliminate ambiguity.

Sometimes the correct conclusion is that dominance itself remains unresolved.

These cases test whether models can distinguish between:

No dominant constraint exists

and

The dominant constraint has not yet been identified.

This distinction separates uncertainty recognition from attribution failure.

14.5 Contested Dominance

Contested dominance occurs when multiple constraints actively compete for control of the trajectory.

Unlike balanced constraints, contested dominance involves competition rather than simple equivalence.

Several mechanisms remain plausible.

The system has not yet resolved which mechanism will ultimately govern future behavior.

Examples include:

- Repair decline competing with signaling degradation
- Compensation exhaustion competing with metabolic collapse
- Multiple active pathways exerting comparable influence

In such situations:

- Several explanations remain structurally viable
- Small perturbations may alter dominance relationships
- Future trajectories remain highly sensitive to change

Contested dominance represents one of the richest sources of structural ambiguity within the framework.

The ambiguity arises not because influence is equal, but because influence remains unstable.

The system is effectively in the process of determining its future dominant state.

Mechanism attribution must therefore remain probabilistic rather than absolute.

These cases are particularly valuable because they resist deterministic attribution and force models to reason about competing structural explanations rather than selecting the most obvious signal.

Temporal Ambiguity

14.6 Dominance Switching

Dominance relationships are not necessarily static.

A constraint that is secondary today may become dominant tomorrow.

A governing mechanism may lose influence as conditions evolve.

Compensatory processes may temporarily suppress one constraint while amplifying another.

Constraint Geometry refers to these transitions as **dominance switching**.

For example:

Early Phase

Signal degradation dominates system behavior.

Intermediate Phase

Compensation masks signaling failure.

Late Phase

Repair capacity becomes exhausted and assumes dominance.

The observable outcome may remain similar throughout this sequence.

The governing mechanism does not.

Dominance switching introduces a temporal dimension into mechanism attribution.

The benchmark question expands from:

Which constraint governs the trajectory?

to:

Which constraint governs the trajectory now?

These benchmark cases evaluate whether models can reason about evolving structural conditions rather than static explanations.

They also highlight an important principle:

Mechanism attribution is often a time-dependent problem rather than a static classification problem.

Compensation Ambiguity

14.7 Compensation-Induced Misclassification

Compensation creates a unique dominance failure mode.

A compensatory process may successfully preserve observable function despite growing instability.

As a result, the dominant constraint becomes masked by the very mechanism attempting to manage it.

For example:

A declining repair process may remain hidden because compensatory pathways continue to maintain apparent stability.

Observed behavior suggests:

Stable system

Underlying structure indicates:

Approaching repair collapse

The model therefore attributes dominance incorrectly.

Not because the dominant constraint is absent.

But because compensation obscures its influence.

Compensation-induced misclassification represents one of the clearest manifestations of outcome–mechanism decoupling within the framework.

Observable outcomes suggest stability while the governing mechanism indicates instability.

This failure mode is particularly important because it directly challenges outcome-based reasoning.

The observable outcome appears stable.

The structural explanation indicates deterioration.

Compensation-induced misclassification therefore provides a direct test of whether a model can reason beyond outcomes and identify the mechanisms operating beneath them.

14.8 Failure Modes as Benchmark Assets

Traditional benchmark design often treats ambiguity as noise to be removed.

Constraint Geometry treats many forms of ambiguity as benchmark assets.

The dominance failure modes introduced in this section are valuable precisely because they create situations in which mechanism attribution becomes non-trivial.

Each failure mode preserves structural ambiguity in a different way.

Failure Mode	Ambiguity Category	Source of Ambiguity	Primary Benchmark Risk

False Dominance	Visibility	Apparent and actual dominance diverge	Signal ranking
Hidden Dominance	Visibility	Governing mechanism is partially unobserved	Observational incompleter
Balanced Constraints	Competition	No dominant mechanism exists	Forced attribution
Contested Dominance	Competition	Multiple mechanisms compete for control	Premature certainty
Dominance Switching	Temporal	Dominance changes through time	Static reasoning
Compensation-Induced Misclassification	Compensation	Compensation obscures dominance	Outcome substitution

These conditions make dominance attribution difficult without rendering it impossible.

They preserve mechanism-attribution signal while preventing the benchmark from collapsing into simple outcome recognition or signal ranking.

These failure modes are particularly valuable because they create situations in which simple signal-ranking strategies fail, thereby reducing opportunities for capability substitution. Successful performance increasingly requires reasoning about governing influence rather than merely identifying the most visible signal.

From a benchmark-design perspective, dominance failure modes function as controllable generators of structural ambiguity.

14.9 Implications for Benchmark Design

The existence of dominance failure modes reinforces a central principle of Constraint Geometry:

Mechanism attribution becomes meaningful only when alternative explanations remain plausible.

If dominance is always obvious, mechanism attribution collapses into signal detection.

If dominance is always hidden, attribution becomes impossible.

Effective benchmark design therefore requires a balance between determinism and ambiguity.

The failure modes introduced in this section provide a structured way to achieve that balance.

They create benchmark tasks in which:

- Multiple explanations remain viable
- Structural ambiguity is preserved
- Outcome recognition becomes insufficient
- Mechanism attribution remains measurable

In this sense, dominance failure modes function as a benchmark-design resource rather than merely a source of error.

They provide the conditions under which mechanism attribution can be evaluated rigorously.

Traditional benchmarks often attempt to eliminate ambiguity.

Constraint Geometry deliberately preserves it.

The objective is not to create tasks in which the correct explanation is immediately obvious.

The objective is to create tasks in which multiple explanations remain plausible, yet one explanation remains structurally superior.

Dominance failure modes provide precisely these conditions.

They transform ambiguity from a source of benchmark noise into a measurable signal of mechanism-attribution performance.

The chapter therefore completes an important transition within the framework.

Structural ambiguity is no longer treated as a problem to be removed.

It becomes a benchmark resource to be engineered, controlled, and measured.

The next section examines how this philosophy differs from existing benchmark paradigms and what becomes visible when evaluation shifts from state recognition toward structural explanation.

Part VII — Comparative Benchmark Analysis

15. Existing Biomedical Benchmarks

The purpose of this paper is not to argue that existing biomedical benchmarks are inadequate.

Contemporary biomedical benchmarks have driven substantial progress in diagnostic prediction, risk estimation, prognosis, treatment selection, and clinical decision support. Many have become important evaluation standards for both traditional machine-learning systems and modern foundation models.

Constraint Geometry builds upon this progress rather than replacing it.

The central question is therefore not whether existing benchmarks are useful.

The central question is:

What capabilities were they designed to measure?

Most biomedical benchmarks were developed to evaluate state recognition, outcome prediction, or risk estimation.

Constraint Geometry proposes an additional evaluation target:

mechanism attribution.

This distinction motivates the comparative analysis presented in this section.

The goal is not to establish superiority.

The goal is to identify differences in evaluation focus.

The strongest argument is not that existing benchmarks are deficient.

The strongest argument is that they were built to measure outcomes, whereas Constraint Geometry is designed to measure explanations.

15.1 Evaluation Targets

Benchmark categories differ in content.

They also differ in what they treat as the primary object of evaluation.

Most biomedical benchmarks evaluate:

- States
- Outcomes
- Risks
- Responses

Constraint Geometry evaluates:

- Structural explanations
- Governing constraints
- Mechanism attribution

The purpose of this comparison is therefore not to compare domains.

It is to compare evaluation targets.

The distinction can be summarized simply:

Outcomes describe what happened.

Explanations describe why it happened.

Traditional biomedical benchmarks primarily evaluate outcomes.

Constraint Geometry introduces explanations as explicit benchmark targets.

15.2 Diagnosis Benchmarks

Diagnosis benchmarks evaluate whether a model can correctly identify the state of a system.

Examples include:

- Disease classification
- Differential diagnosis
- Image-based diagnosis
- Pathology classification
- Clinical case interpretation

The benchmark question is typically:

What condition is present?

Examples include:

- Is disease present?
- Which diagnosis best fits the evidence?
- Which pathology category applies?

These tasks evaluate recognition and classification.

They measure whether a model can correctly map observations to diagnostic labels.

What Diagnosis Benchmarks Measure Well

Diagnosis benchmarks are highly effective at evaluating:

- Pattern recognition
- Classification accuracy
- Differential diagnosis
- Clinical feature integration
- State identification

These represent important and clinically valuable capabilities.

What Diagnosis Benchmarks Do Not Explicitly Measure

Diagnosis benchmarks typically do not require models to distinguish among competing structural explanations that produce similar diagnostic states.

Two patients may receive the same diagnosis despite differing substantially in:

- Governing constraints
- Compensation status
- Dominant mechanisms
- Failure pathways

The benchmark evaluates whether the diagnosis is correct.

It does not necessarily evaluate whether the mechanism responsible for the diagnosis has been correctly identified.

15.3 Prognosis Benchmarks

Prognosis benchmarks evaluate future outcomes.

The benchmark question becomes:

What is likely to happen next?

Examples include:

- Disease progression
- Relapse prediction
- Functional decline
- Recovery probability
- Future risk estimation

These tasks shift evaluation from present-state recognition toward future-state prediction.

What Prognosis Benchmarks Measure Well

Prognosis benchmarks effectively evaluate:

- Predictive accuracy
- Risk estimation
- Temporal forecasting
- Longitudinal prediction
- Outcome modeling

These capabilities are central to modern predictive medicine.

What Prognosis Benchmarks Do Not Explicitly Measure

Accurate prognosis does not necessarily reveal the mechanism responsible for the predicted trajectory.

A model may correctly predict progression while relying on:

- Statistical associations
- Historical patterns
- Outcome correlations

without identifying the governing mechanism responsible for the future outcome.

The benchmark evaluates:

Will progression occur?

It does not necessarily evaluate:

Why will progression occur?

15.4 Survival Prediction Benchmarks

Survival prediction benchmarks evaluate the probability or timing of future survival outcomes.

Examples include:

- Overall survival prediction
- Progression-free survival
- Event-free survival
- Mortality prediction

The benchmark objective is typically:

How long is survival expected to persist?

What Survival Benchmarks Measure Well

These benchmarks are effective at evaluating:

- Outcome prediction
- Risk stratification
- Temporal estimation
- Hazard modeling
- Longitudinal forecasting

Survival prediction has become one of the most important applications of predictive modeling in medicine.

What Survival Benchmarks Do Not Explicitly Measure

Survival outcomes compress substantial structural information.

Different mechanisms may produce similar survival trajectories.

Different governing constraints may produce similar mortality risks.

Two patients may receive identical survival predictions while being governed by different structural conditions.

The benchmark evaluates:

What outcome is likely?

It generally does not evaluate:

Which mechanism governs that outcome?

15.5 Treatment Response Benchmarks

Treatment-response benchmarks evaluate whether a model can predict the effect of an intervention.

Examples include:

- Drug-response prediction
- Treatment selection
- Therapy stratification
- Precision-medicine benchmarks
- Intervention-outcome prediction

The benchmark question becomes:

Will this intervention succeed?

What Treatment-Response Benchmarks Measure Well

Treatment-response benchmarks effectively evaluate:

- Intervention forecasting
- Response classification
- Therapeutic stratification
- Treatment prioritization
- Outcome prediction under intervention

These capabilities are often directly relevant to clinical decision-making.

What Treatment-Response Benchmarks Do Not Explicitly Measure

Successful response prediction does not necessarily imply identification of the mechanism through which the response occurs.

A model may correctly predict that a treatment will succeed without identifying:

- The governing constraint
- The dominant mechanism
- The compensatory processes involved
- The structural explanation responsible for the response

The benchmark evaluates:

Will the treatment work?

It does not necessarily evaluate:

Why does the treatment work?

15.6 The Common Evaluation Pattern

Despite substantial differences among diagnosis, prognosis, survival, and treatment-response benchmarks, they share a common evaluation structure.

The benchmark target is typically an outcome.

Examples include:

- Diagnosis
- Risk category
- Progression
- Survival
- Response

Outcomes serve as compressed representations of system behavior.

Benchmark performance is evaluated by comparing predicted outcomes against observed outcomes.

This architecture has proven highly successful because outcomes are often observable, measurable, and clinically relevant.

However, the structural explanation responsible for those outcomes usually remains outside the benchmark target itself.

This observation does not imply a flaw.

It reflects the design objectives of these benchmarks.

They were created to evaluate outcomes.

Constraint Geometry is designed to evaluate explanations.

15.7 Outcome Targets vs Mechanism Targets

The distinction can be summarized as follows.

Benchmark Type	Primary Evaluation Target	Core Question
Diagnosis Benchmark	State	What condition is present?
Prognosis Benchmark	Future Outcome	What will happen?
Survival Benchmark	Survival Outcome	How long will function persist?
Treatment-Response Benchmark	Intervention Outcome	Will the intervention succeed?
Constraint Geometry Benchmark	Structural Explanation	Why did it happen?

Importantly, these categories are not mutually exclusive.

A benchmark may evaluate both outcomes and mechanisms.

Indeed, the dual-target architecture introduced earlier was designed precisely for this purpose.

The difference lies in what is treated as the primary object of evaluation.

Traditional biomedical benchmarks primarily evaluate outcomes.

Constraint Geometry evaluates both outcomes and the mechanisms believed to generate them.

15.8 What Existing Benchmarks Measure Well

Existing biomedical benchmarks have been extraordinarily successful at measuring:

- Classification
- Recognition
- Forecasting
- Risk estimation
- Outcome prediction
- Treatment-response prediction

These capabilities remain essential.

Constraint Geometry does not propose replacing them.

In many settings they may be entirely sufficient.

The framework introduced in this paper should therefore be understood as complementary rather than competitive.

15.9 What Existing Benchmarks Do Not Explicitly Measure

The central observation motivating this work is that mechanism attribution rarely appears as an explicit benchmark target.

Most benchmark tasks do not require models to distinguish among competing structural explanations that remain consistent with similar observations.

Most benchmark tasks do not evaluate:

- Governing constraints
- Dominance relationships
- Compensation status
- Readability failure
- Alignment failure
- Latency failure
- Structural ambiguity

These concepts may influence model performance.

They are not typically the object of evaluation.

Constraint Geometry proposes making them explicit benchmark targets.

The framework therefore introduces a different question into benchmark design:

Can a model correctly identify the structural explanation responsible for an observed trajectory?

This question does not replace diagnosis, prognosis, survival prediction, or treatment-response prediction.

It supplements them.

15.10 Comparative Interpretation

The distinction between outcome-oriented benchmarks and mechanism-attribution benchmarks should not be interpreted as a distinction between good benchmarks and bad benchmarks.

It is a distinction between evaluation objectives.

Traditional biomedical benchmarks ask:

What happened?

or

What is likely to happen?

Constraint Geometry asks:

Why did it happen?

More specifically:

Which structural explanation best accounts for the observed trajectory?

The framework therefore extends benchmark evaluation from outcomes alone toward explicit evaluation of explanations.

Whether explanation and prediction ultimately prove separable remains an empirical question.

Constraint Geometry does not assume the answer.

It provides a benchmark architecture in which the relationship between outcome recognition and mechanism attribution can be measured directly.

15.11 Conclusion

Existing biomedical benchmarks have been extraordinarily successful because they evaluate outcomes that matter.

Constraint Geometry does not challenge that objective.

Instead, it proposes an additional evaluation target.

Traditional benchmarks ask:

What happened?

Constraint Geometry asks:

Why did it happen?

More specifically:

Which structural explanation best accounts for what happened?

The framework should therefore be understood as complementary rather than competitive.

It extends benchmark evaluation from outcomes toward explanations, and from prediction alone toward explicit evaluation of mechanism attribution.

The following section examines what becomes visible when benchmark evaluation expands beyond outcome prediction and begins measuring structural explanations directly.

I would reflow the back half of the chapter so it reads less like a collection of additions and more like a progression from **new evaluation target → new measurement architecture → new benchmark capability**.

The biggest structural change is:

- Move **Capability Isolation** immediately after **Outcome–Mechanism Decoupling**
- Move **Structural Alignment Accuracy** after Capability Isolation
- Tighten **Governing Constraint Identification**
- Elevate **Failure Reconstruction** into a major contribution rather than a supporting concept

The resulting chapter becomes:

---

16. What Constraint Geometry Adds

The preceding section examined the primary evaluation targets of existing biomedical benchmarks and argued that most were designed to evaluate states, risks, outcomes, or intervention responses.

Constraint Geometry does not seek to replace these benchmark categories.

Instead, it introduces an additional evaluation target:

mechanism attribution.

The contribution of the framework is therefore not a new prediction task.

It is a new evaluation perspective.

The central question is not whether models can predict outcomes.

The central question is whether models can identify the structural explanations responsible for those outcomes.

Viewed in this way, Constraint Geometry does not primarily change what benchmarks evaluate.

It changes what benchmark performance means.

A correct prediction is no longer the only object of interest.

The structural explanation underlying that prediction becomes an object of evaluation as well.

This section summarizes the principal additions introduced by the Constraint Geometry framework and clarifies how they differ from conventional benchmark architectures.

16.1 A Shift in Evaluation Target

The most fundamental contribution of Constraint Geometry is a shift in what is treated as the primary object of evaluation.

Most benchmark categories evaluate observable states or outcomes.

Constraint Geometry evaluates structural explanations.

Benchmark Type	Primary Evaluation Target
Diagnosis	State
Risk Prediction	Probability
Prognosis	Outcome
Survival Prediction	Outcome

Treatment Response	Outcome
Constraint Geometry	Structural Explanation

This distinction does not imply that explanations are more important than outcomes.

Rather, it reflects a different evaluation objective.

Traditional benchmarks primarily ask:

What happened?

Constraint Geometry asks:

What structural explanation best accounts for what happened?

This shift in evaluation target serves as the foundation for every subsequent component of the framework.

16.2 Mechanism Attribution as an Explicit Benchmark Target

Many existing benchmarks contain underlying mechanisms.

Few evaluate mechanism attribution directly.

Mechanisms often remain implicit.

They influence benchmark performance without becoming benchmark targets themselves.

Constraint Geometry makes mechanism attribution explicit.

The benchmark objective is no longer limited to predicting outcomes.

It includes identifying:

- Governing constraints
- Dominant mechanisms
- Compensation states
- Readability failures
- Alignment failures
- Latency failures
- Competing structural explanations

Mechanism attribution therefore becomes a measurable component of benchmark performance rather than an inferred consequence of outcome prediction.

This represents the central conceptual contribution of the framework.

Explanations become benchmark targets.

16.3 From Outcome Evaluation to Explanation Evaluation

Traditional benchmark architectures evaluate outcomes.

Constraint Geometry evaluates both outcomes and explanations.

Outcomes describe observable system states.

Explanations describe the structural conditions responsible for generating those states.

The framework therefore introduces a new benchmark question:

Which explanation best accounts for the observed trajectory?

Importantly, this question does not replace outcome prediction.

Instead, it supplements outcome prediction with a second layer of evaluation.

The result is a benchmark architecture capable of evaluating not only whether a prediction is correct, but whether the structural explanation associated with that prediction is correct as well.

16.4 Dual-Target Evaluation

The introduction of explanation as a benchmark target naturally leads to dual-target evaluation.

Traditional benchmarks typically evaluate a single target:

Outcome Correctness

Constraint Geometry evaluates two targets simultaneously:

Target A

Outcome Recognition

Target B

Mechanism Attribution

This architecture allows benchmark performance to be decomposed into:

- Outcome performance
- Mechanism performance
- Outcome–mechanism divergence

The relationship between these targets becomes an empirical result rather than an assumption.

If outcome prediction and mechanism attribution remain tightly coupled, the benchmark will reveal that.

If meaningful separation exists, the benchmark will reveal that as well.

Dual-target evaluation therefore transforms a conceptual distinction into a measurable one.

16.5 Structural Ambiguity as a Benchmark Design Principle

Mechanism attribution becomes meaningful only when multiple explanations remain plausible.

For this reason, Constraint Geometry introduces structural ambiguity as a benchmark-design principle.

Traditional benchmark design often attempts to eliminate ambiguity.

Constraint Geometry deliberately preserves certain forms of ambiguity because ambiguity carries mechanism-attribution signal.

Examples include:

- Different mechanisms producing similar outcomes
- Similar mechanisms producing different outcomes
- Compensation masking instability
- Contested dominance relationships
- Partial observability
- Delayed effects

These conditions prevent benchmark performance from collapsing into simple prediction, signal ranking, or shortcut detection.

Structural ambiguity therefore functions as a design resource rather than a source of benchmark noise.

16.6 Outcome–Mechanism Decoupling

A central contribution of the framework is the recognition that outcomes and mechanisms represent different informational targets.

Outcomes describe observable system states.

Mechanisms describe the structural conditions responsible for those states.

Constraint Geometry therefore introduces outcome–mechanism decoupling as a measurable benchmark property.

The framework explicitly investigates situations in which:

- Outcome prediction succeeds while mechanism attribution fails
- Mechanism attribution succeeds while outcome prediction remains uncertain
- Different mechanisms produce identical outcomes
- Similar mechanisms produce different outcomes

The dual-target architecture makes such divergences measurable.

As a result, the relationship between prediction and explanation becomes an empirical question rather than an assumption.

16.7 Mechanism Attribution as Capability Isolation

A central motivation for Constraint Geometry is the possibility that outcome prediction and mechanism attribution may not represent identical benchmark targets.

Traditional benchmarks often aggregate multiple capabilities into a single outcome metric.

As a result, it may be difficult to determine which capabilities contribute to successful performance.

Constraint Geometry seeks to isolate mechanism attribution as an explicit evaluation target.

The framework does not assume that mechanism attribution constitutes a distinct capability.

Rather, it creates the conditions under which that question can be investigated empirically.

By evaluating outcomes and mechanisms separately, the benchmark architecture makes it possible to observe situations in which:

- Outcome prediction succeeds while mechanism attribution fails
- Mechanism attribution succeeds while outcome prediction remains uncertain

If such divergences occur systematically, they provide evidence that the two targets are not being solved identically.

Mechanism attribution can therefore be understood as an attempt to isolate a potentially distinct form of reasoning for benchmark evaluation.

16.8 Structural Alignment Accuracy

Traditional benchmark metrics evaluate outcome correctness.

Constraint Geometry introduces an additional mechanism-oriented metric:

Structural Alignment Accuracy (SAA)

SAA measures the degree of correspondence between:

- The explanation inferred by the model
- The mechanism responsible for generating the benchmark trajectory

Conceptually, SAA plays the same role for explanations that classification accuracy plays for outcomes.

Outcome Accuracy evaluates:

Was the outcome predicted correctly?

Structural Alignment Accuracy evaluates:

Was the structural explanation identified correctly?

The introduction of SAA transforms mechanism attribution from a conceptual objective into a measurable benchmark variable.

16.9 Governing Constraint Identification

Many complex systems contain multiple active constraints simultaneously.

Some constraints influence behavior.

Others determine trajectory.

Constraint Geometry introduces **governing constraint identification** as the task of determining which active constraint most strongly governs future system behavior.

This distinction is important because mechanism attribution often involves competing explanations rather than isolated mechanisms.

A system may exhibit:

- Repair limitations
- Signaling degradation
- Compensation activity
- Resource constraints

at the same time.

The benchmark challenge is not merely to identify these constraints.

It is to determine which constraint exerts the greatest influence over the future trajectory of the system.

In this sense, governing constraint identification can be understood as **influence ranking under structural ambiguity**.

The objective is not to identify everything that matters.

The objective is to identify what matters most.

Constraint Geometry formalizes this task through dominance attribution and treats it as an explicit benchmark target.

16.10 Failure Reconstruction

Most benchmark architectures evaluate outcomes.

Constraint Geometry extends evaluation toward reconstruction.

The framework treats mechanism attribution as a form of failure reconstruction.

Prediction asks:

What happened?

Explanation asks:

Why did it happen?

Failure reconstruction asks:

What sequence of structural conditions produced what happened?

This distinction is important because many real-world investigations focus on reconstruction rather than prediction.

Examples include:

- Clinical case review
- Root-cause analysis
- Accident investigation
- System failure analysis
- Operational postmortems

In each case, the objective is not merely to identify the outcome.

The objective is to reconstruct the structural pathway that generated the outcome.

Constraint Geometry introduces benchmark tasks designed to evaluate this form of reasoning directly.

Mechanism attribution, dominance attribution, and structural explanation can therefore be understood as components of benchmark-level failure reconstruction.

This extends evaluation beyond prediction toward assessment of whether models can reconstruct the mechanisms responsible for observed trajectories.

16.11 A Different Interpretation of Benchmark Performance

Taken together, these additions change the interpretation of benchmark success.

Traditional benchmark performance is typically interpreted as evidence that a model can recognize or predict outcomes.

Constraint Geometry introduces an additional possibility.

High benchmark performance may also require identifying the structural explanations responsible for those outcomes.

The framework therefore shifts evaluation from:

Can the model predict correctly?

toward:

Can the model predict for structurally appropriate reasons?

This distinction represents the practical significance of mechanism attribution as a benchmark target.

16.12 Summary

Constraint Geometry introduces seven principal additions to benchmark evaluation:

1. Structural explanation as an explicit evaluation target.
2. Mechanism attribution as a measurable benchmark task.
3. Dual-target evaluation of outcomes and mechanisms.
4. Structural ambiguity as a benchmark-design principle.
5. Outcome–mechanism decoupling as a measurable property.
6. Capability isolation through explicit mechanism evaluation.
7. Failure reconstruction as a benchmark objective.

Together these additions extend benchmark evaluation from outcomes toward explanations, and from prediction alone toward explicit evaluation of mechanism attribution.

The framework does not propose a new predictive model, biological theory, or clinical system.

It proposes a benchmark architecture designed to investigate whether mechanism attribution represents a measurable benchmark target distinct from state recognition and outcome prediction.

Whether that distinction ultimately corresponds to a distinct capability remains an empirical question.

Constraint Geometry does not assume the answer.

It provides a benchmark environment in which the question can be measured.

17. Why This Difference Matters

The preceding sections argued that outcome prediction and mechanism attribution represent distinct evaluation targets. Constraint Geometry was introduced as a benchmark framework designed to investigate that distinction empirically.

A natural question follows:

Why does the distinction matter?

If two benchmark approaches produce similar predictive performance, does it matter whether one evaluates outcomes while the other evaluates mechanisms?

The answer depends on what benchmark performance is intended to represent.

If the objective is prediction alone, outcome accuracy may be sufficient.

If the objective includes explanation, intervention selection, failure reconstruction, governing constraint identification, or reasoning under structural ambiguity, then outcome prediction may provide only a partial picture.

The examples in this section illustrate why mechanism attribution can remain important even when outcomes are known.

These examples are intentionally simplified.

Their purpose is not to model real clinical situations but to demonstrate how outcome recognition and structural explanation can diverge.

More fundamentally, they illustrate a central idea underlying the Constraint Geometry framework:

Outcomes compress information.

Explanations recover information.

Outcome labels summarize trajectories into observable states.

Mechanism attribution attempts to recover the structural distinctions that those labels eliminate.

The importance of mechanism attribution therefore depends on whether those structural distinctions matter for understanding, intervention, or explanation.

17.1 Outcome Compression and Explanation Recovery

Many benchmark tasks evaluate outcomes because outcomes are observable, measurable, and operationally useful.

This design choice is both practical and effective.

However, outcomes also compress information.

A trajectory containing multiple interacting constraints, compensatory processes, dominance relationships, and structural transitions may ultimately be represented by a single label:

- Failure
- Recovery
- Progression
- Stability

These labels capture what happened.

They do not necessarily preserve information about why it happened.

Mechanism attribution can therefore be understood as a form of explanation recovery.

The objective is to reconstruct structural distinctions that are lost when trajectories are compressed into outcomes.

The examples that follow illustrate several ways in which this compression can occur.

17.2 Case Study 1: Same Outcome, Different Mechanisms

Consider two systems.

System A

Outcome:

Failure

Dominant Mechanism:

Repair Capacity Collapse

The system receives adequate information and sufficient resources.

The failure occurs because repair demand gradually exceeds repair capacity.

Compensation eventually becomes exhausted.

The system fails.

System B

Outcome:

Failure

Dominant Mechanism:

Signal Readability Collapse

Repair capacity remains largely intact.

Resources remain available.

The failure occurs because critical signals become increasingly unreadable.

Adaptive responses are therefore misdirected.

The system fails.

Outcome Perspective

Both systems belong to the same category:

Failure

A benchmark evaluating outcomes alone treats the cases as equivalent.

Mechanism Perspective

The cases are fundamentally different.

System A is governed by insufficient repair capacity.

System B is governed by information failure.

The observable outcome is identical.

The governing structure is not.

Information Perspective

Two distinct structural explanations have been compressed into a single outcome category.

Mechanism attribution recovers the distinction.

Benchmark Implication

A model may correctly predict failure in both cases while assigning the same explanation to each.

Outcome prediction succeeds.

Mechanism attribution fails.

Traditional benchmarks often cannot detect this distinction because explanation quality is not evaluated directly.

Constraint Geometry can.

17.3 Case Study 2: Same Mechanism, Different Outcomes

Now consider a different situation.

System C

Dominant Mechanism:

Compensation Exhaustion

Outcome:

Failure

Compensation collapses before recovery can occur.

The system becomes unstable.

Failure follows.

System D

Dominant Mechanism:

Compensation Exhaustion

Outcome:

Recovery

Compensation declines, but intervention occurs before critical thresholds are crossed.

The system recovers.

Outcome Perspective

The systems belong to different categories:

- Failure
- Recovery

A benchmark focused on outcomes treats them as distinct cases.

Mechanism Perspective

The governing mechanism is identical.

Both systems are governed by compensation exhaustion.

The difference arises from timing, intervention, and trajectory evolution.

Information Perspective

A single mechanism has been fragmented across multiple outcome categories.

Mechanistic similarity becomes invisible.

Benchmark Implication

Outcome categories alone obscure structural commonality.

A model that recognizes the shared governing mechanism demonstrates information not captured by outcome prediction.

Constraint Geometry explicitly evaluates this capability.

17.4 Case Study 3: Compensation Masks Instability

Compensation creates one of the most important sources of outcome–mechanism divergence.

Consider two systems.

System E

Observed State:

Stable

Compensation:

Minimal

Underlying Stability:

Genuinely Stable

System F

Observed State:

Stable

Compensation:

Extreme

Underlying Stability:

Approaching Collapse

Compensatory processes preserve observable function despite rapidly increasing structural pressure.

The system appears healthy.

The trajectory indicates otherwise.

Outcome Perspective

Both systems appear stable.

A benchmark evaluating observable state alone may classify both correctly.

Mechanism Perspective

The systems are radically different.

System E is stable because constraints remain aligned.

System F is stable because compensation temporarily conceals instability.

The observable state is the same.

The structural condition is not.

Information Perspective

Compensation creates a separation between appearance and governing structure.

The outcome remains unchanged while the underlying mechanism changes dramatically.

Benchmark Implication

Compensation creates situations in which observable outcomes diverge from underlying structure.

Mechanism attribution becomes necessary precisely because outcome recognition is insufficient.

17.5 Case Study 4: Partial Observability

Mechanism attribution becomes even more important when information is incomplete.

Consider two systems.

System G

Observed Information:

Partial

Missing Information:

Non-critical

Outcome:

Stable

System H

Observed Information:

Partial

Missing Information:

Critical

Outcome:

Unstable

Both systems appear similar from the perspective of available observations.

Both contain missing information.

Only one contains missing information relevant to the governing mechanism.

Outcome Perspective

The systems differ only in their outcomes.

A predictive benchmark evaluates whether the outcome was identified correctly.

Mechanism Perspective

The crucial distinction concerns the role of the missing information.

In System G, missing information does not govern behavior.

In System H, missing information conceals the dominant constraint.

Information Perspective

The benchmark challenge is not simply whether information is missing.

It is whether the missing information matters structurally.

Benchmark Implication

Partial observability introduces uncertainty that cannot always be resolved through additional prediction.

Mechanism attribution requires reasoning about the structural significance of missing information.

Traditional outcome-oriented benchmarks rarely evaluate this capability directly.

17.6 What Outcome-Oriented Evaluation Misses

The preceding examples illustrate a broader point.

Several explanatory failures can remain invisible within outcome-focused benchmark architectures.

Examples include:

Correct Prediction, Incorrect Explanation

The model predicts the correct outcome but assigns the wrong mechanism.

False Dominance Attribution

The model identifies a visible constraint rather than the governing constraint.

Compensation Blindness

The model interprets compensated stability as genuine stability.

Readability Misclassification

The model confuses degraded information with missing information.

Latency Misattribution

The model attributes failure to burden rather than timing.

Equifinality Collapse

The model treats distinct mechanisms as equivalent because they produce similar outcomes.

In each case, outcome prediction may remain correct.

Mechanism attribution does not.

Traditional benchmarks often register these cases as successes.

Constraint Geometry registers them as explanatory failures.

17.7 Prediction, Explanation, and Failure Reconstruction

The purpose of these examples is not to diminish the importance of prediction.

Prediction remains one of the most valuable capabilities in applied machine learning.

Constraint Geometry does not propose replacing predictive evaluation.

Instead, it introduces two additional evaluation questions.

Explanation asks:

Why did it happen?

Failure reconstruction asks:

What sequence of structural conditions produced what happened?

This distinction is important because many real-world investigations focus on reconstruction rather than prediction.

Clinical case reviews, accident investigations, root-cause analyses, and operational postmortems rarely stop after identifying an outcome.

They attempt to reconstruct the pathway that produced it.

Constraint Geometry extends benchmark evaluation toward this form of reasoning.

Mechanism attribution can therefore be understood not only as explanation, but also as benchmark-level failure reconstruction.

17.8 Why the Difference Matters

The distinction between outcome prediction and mechanism attribution ultimately matters because interventions act on mechanisms rather than outcomes.

This principle lies at the center of the framework:

> **Interventions do not operate on outcomes.**
> **They operate on mechanisms.**

Outcomes describe what happened.

Mechanisms describe what produced what happened.

If different mechanisms produce similar outcomes, intervention decisions may depend on identifying the correct explanation rather than merely recognizing the outcome.

Constraint Geometry does not claim that mechanism attribution is always necessary.

Nor does it claim that mechanism attribution represents a distinct capability.

Those remain empirical questions.

The framework advances a more modest proposal:

If mechanism attribution matters, benchmark evaluation should be able to measure it.

The examples in this section illustrate why outcome recognition alone may be insufficient to evaluate that capability.

They provide a concrete demonstration of the central argument developed throughout this paper:

Knowing what happened is not always the same as knowing why it happened.

Constraint Geometry is designed to investigate that distinction.

19. Scope

The preceding sections introduced Constraint Geometry as a framework for evaluating mechanism attribution, structural explanation, and governing constraint identification within synthetic benchmark environments.

As the framework intersects with topics such as oncology, systems biology, causal reasoning, and artificial intelligence evaluation, it is important to define the scope of its contribution precisely.

The purpose of this work is not to advance a new scientific theory of disease, propose a clinical framework, or establish novel biological mechanisms.

Its contribution is substantially narrower.

The paper is best understood as a contribution to **benchmark methodology** rather than to biology, medicine, or scientific theory.

More specifically, it concerns how benchmark environments can be constructed to evaluate mechanism attribution, structural explanation, and governing constraint identification under controlled conditions.

This distinction is important because benchmark architectures and scientific theories address different questions.

Scientific theories seek to explain how real systems operate.

Benchmark architectures seek to evaluate the capabilities of models operating within defined environments.

Constraint Geometry belongs to the latter category.

19.1 What This Work Contributes

The primary contribution of this work is a benchmark architecture for evaluating mechanism attribution.

The framework introduces a benchmark category in which models are evaluated not only on outcome recognition but also on their ability to identify the structural explanations responsible for observed trajectories.

Several specific contributions follow from this objective.

Benchmark Architecture

Constraint Geometry introduces a benchmark architecture centered on structural explanation rather than outcome prediction alone.

The framework defines benchmark tasks in which multiple structural explanations may remain consistent with similar observations.

This architecture enables direct evaluation of mechanism attribution under conditions of structural ambiguity.

Benchmark Design Methodology

The paper introduces a set of benchmark-design principles intended to preserve mechanism-attribution signal.

These principles include:

- Structural ambiguity
- Outcome–mechanism separation
- Compensation-aware design
- Dominance ambiguity
- Partial observability
- Anti-shortcut controls

Together these principles form a methodology for constructing benchmark environments that resist reduction to simple prediction tasks.

Structural Reasoning Evaluation

The framework introduces structural reasoning as an explicit object of benchmark evaluation.

Rather than asking only whether a model can predict an outcome, the framework evaluates whether it can identify the structural explanation most consistent with the available evidence.

This contribution is evaluative rather than explanatory.

The framework measures structural attribution.

It does not claim to discover structural truth.

Dual-Target Evaluation

The framework introduces a dual-target architecture in which outcome recognition and mechanism attribution are evaluated separately.

This architecture allows benchmark performance to be decomposed into:

- Outcome performance
- Mechanism-attribution performance
- Outcome–mechanism divergence

The relationship between these targets becomes an empirical result rather than an assumption.

19.2 What This Work Does Not Contribute

The framework should not be interpreted as making contributions beyond benchmark evaluation.

Several categories of contribution fall explicitly outside the scope of this paper.

Biomedical Discovery

This work does not establish new biomedical findings.

It does not identify disease mechanisms.

It does not generate biological discoveries.

It does not provide evidence regarding the true causes of disease.

The benchmark environments used throughout this paper are synthetic and exist solely for evaluation purposes.

Oncology Theory

The framework is not a theory of cancer.

The oncology-inspired benchmark family is used because it provides a useful environment for constructing mechanism-attribution tasks.

The benchmark suite does not attempt to explain tumor biology, disease progression, treatment response, or cancer causation.

The significance of oncology within this framework is therefore not biomedical.

It is architectural.

Clinical Decision Support

Constraint Geometry is not a clinical decision-support framework.

The benchmark outputs are not intended to guide diagnosis, prognosis, treatment selection, or patient management.

Performance within synthetic benchmark environments should not be interpreted as evidence of clinical competence.

The framework evaluates benchmark behavior rather than clinical utility.

Disease Modeling

The framework is not a disease model.

The benchmark environments are designed to generate mechanism-attribution tasks rather than to represent the dynamics of any real biological system.

Their purpose is evaluative rather than descriptive.

19.3 Synthetic Attribution Versus Structural Discovery

An additional scope distinction concerns the relationship between synthetic benchmark evaluation and real-world scientific reasoning.

Constraint Geometry evaluates attribution among known competing explanations.

The benchmark designer specifies the mechanisms responsible for generating benchmark trajectories.

This allows structural explanations to be evaluated directly through Structural Alignment Accuracy.

Real-world scientific reasoning often involves a fundamentally different challenge.

The relevant mechanisms may be unknown.

Observations may be incomplete.

Ground truth may be unavailable.

Competing explanations may remain unresolved.

As a result, real-world reasoning frequently requires discovery of previously unknown explanations.

Constraint Geometry does not evaluate this capability.

The framework evaluates attribution among known competing explanations.

It does not evaluate discovery of unknown explanations.

These tasks are related.

They are not the same.

Success within Constraint Geometry benchmarks should therefore not be interpreted as evidence that a model can perform unrestricted structural discovery in real-world domains.

19.4 What This Work Does Not Claim

The framework should also be distinguished from a number of claims that are intentionally left unresolved.

Mechanism Attribution as a Distinct Capability

This paper does not claim that mechanism attribution constitutes a distinct cognitive or computational capability.

The central hypothesis of the benchmark framework is that mechanism attribution can be evaluated separately from outcome recognition.

Whether performance on these targets ultimately proves separable remains an empirical question.

The benchmark architecture is designed to investigate that question rather than assume its answer.

Benchmark Success and Real-World Understanding

This paper does not claim that success on Constraint Geometry benchmarks implies genuine understanding of real-world systems.

Benchmark performance demonstrates success within a defined evaluation environment.

The extent to which such performance transfers to scientific reasoning, engineering reasoning, clinical reasoning, or other real-world domains remains an open question.

Structural Alignment and Causal Truth

This paper does not claim that structural alignment within synthetic benchmark environments corresponds to causal truth in real-world systems.

Structural Alignment Accuracy evaluates correspondence to benchmark-generating mechanisms.

It does not establish the correctness of scientific theories beyond the benchmark itself.

Mechanism Attribution and General Intelligence

This paper does not claim that mechanism attribution constitutes a prerequisite for general intelligence, advanced reasoning, or expert performance.

The framework evaluates a specific benchmark target.

Broader claims regarding intelligence remain outside the scope of the present work.

19.5 Benchmark Evaluation Rather Than Scientific Validation

The central objective of this work is methodological.

The paper asks whether benchmark environments can be designed to evaluate mechanism attribution separately from outcome recognition.

It does not attempt to validate a scientific theory.

It does not attempt to establish causal truth.

It does not attempt to prove that mechanism attribution constitutes a distinct capability.

Instead, it provides a benchmark architecture through which these questions can be investigated empirically.

The framework therefore functions as an evaluation methodology rather than a scientific model.

Its objective is measurement rather than validation.

19.6 Scope Summary

The contribution of Constraint Geometry can be summarized succinctly.

This work contributes:

- A benchmark architecture
- A benchmark-design methodology
- A framework for evaluating structural reasoning
- A framework for evaluating mechanism attribution
- A framework for governing-constraint identification
- A dual-target evaluation architecture

This work does not contribute:

- Biomedical discovery
- Oncology theory
- Clinical decision support
- Disease modeling
- Treatment recommendations
- Causal discovery
- Scientific validation of biological mechanisms

This work does not claim:

- That mechanism attribution is a distinct capability
- That benchmark success implies scientific understanding
- That benchmark success implies clinical competence
- That structural alignment implies causal truth
- That synthetic attribution transfers to real-world discovery

Constraint Geometry is therefore best understood as a benchmark framework.

Its purpose is to evaluate whether models can distinguish among competing structural explanations under controlled conditions.

The framework makes no claim that success on such tasks implies scientific understanding, clinical competence, causal discovery capability, or real-world mechanism attribution.

Those questions remain outside the scope of the present work.

Instead, the framework provides a method for measuring a specific form of benchmark performance and leaves broader questions of transfer, utility, capability structure, and scientific validity to future investigation.

Part IX — Discussion

20. Structural Reasoning as a Benchmark Category

The preceding sections introduced Constraint Geometry as a framework for evaluating mechanism attribution, structural explanation, governing-constraint identification, and outcome–mechanism decoupling. The framework was motivated by a perceived gap in contemporary benchmark ecosystems and operationalized through a family of synthetic benchmark environments designed to preserve structural ambiguity.

The broader question raised by this work extends beyond the specific benchmark suite presented here.

It concerns the organization of benchmark categories themselves.

Most benchmark ecosystems are organized around the outputs models are expected to produce.

Examples include:

- Classification
- Prediction
- Retrieval
- Recommendation
- Generation

These categories have proven enormously valuable because they make particular forms of performance measurable.

Constraint Geometry proposes an alternative perspective.

Benchmark categories may also be organized according to the type of reasoning they require.

Viewed from this perspective, structural reasoning emerges as a potential evaluation category in its own right.

The purpose of this section is not to argue that structural reasoning has already been established as a distinct capability.

Rather, it is to examine whether structural reasoning may represent a useful benchmark category for AI evaluation.

20.1 Benchmark Categories and Evaluation Targets

Benchmark categories do more than organize tasks.

They define what becomes visible to evaluation.

A capability that lacks an explicit benchmark target is difficult to measure systematically.

Classification benchmarks made recognition measurable.

Prediction benchmarks made forecasting measurable.

Robustness benchmarks made resilience measurable.

Retrieval benchmarks made information recovery measurable.

In each case, the benchmark category created a structured way to evaluate a previously diffuse concept.

This observation suggests a broader principle:

Benchmark categories function as measurement instruments.

Their significance lies not only in the tasks they contain but also in the phenomena they make observable.

From this perspective, an important question emerges:

What benchmark category evaluates structural explanation?

The answer is not immediately obvious.

Many existing benchmarks contain mechanisms.

Many contain causal structure.

Many contain interacting processes.

Yet relatively few explicitly evaluate whether a model can identify the structural explanation responsible for observed behavior.

This creates a potential blind spot within benchmark evaluation.

Models may become increasingly effective at predicting outcomes without benchmark ecosystems providing equally direct evidence regarding mechanism attribution.

Constraint Geometry was developed to investigate whether this gap can be addressed through benchmark design.

20.2 A Candidate Benchmark Category

Most benchmark categories can be understood in terms of their primary evaluation targets.

Benchmark Category	Primary Evaluation Target

Classification	States
Prediction	Outcomes
Retrieval	Information
Recommendation	Decisions
Generation	Outputs
Structural Reasoning	Explanations

This comparison is not intended to imply that explanations are more important than outcomes.

Rather, it highlights a difference in what is being evaluated.

Traditional benchmark categories primarily ask:

What is the correct answer?

Structural reasoning asks:

What structural explanation best accounts for the answer?

The distinction mirrors a recurring theme throughout this paper.

Outcomes describe what happened.

Explanations describe why it happened.

If explanation quality represents a meaningful object of evaluation, then structural reasoning may represent a useful benchmark category alongside existing evaluation paradigms.

20.3 From State Recognition to Structural Reasoning

Much of contemporary benchmark evaluation can be understood as a progression from observations to outcomes.

The benchmark task is typically:

Given these observations, what is the correct answer?

Constraint Geometry introduces a different question:

Given these observations, what structural explanation best accounts for them?

This shifts evaluation away from outcomes alone and toward the relationships responsible for generating outcomes.

Structural reasoning, as defined within this framework, concerns the ability to reason about:

- Constraint interactions
- Compensation
- Dominance relationships
- Information degradation
- Latency effects
- Partial observability
- Competing explanations

These concepts differ from simple outcome recognition because they concern relationships among processes rather than outcomes themselves.

The resulting benchmark category is therefore organized around explanation rather than prediction.

20.4 What Becomes Measurable

One of the central motivations for introducing a new benchmark category is that new benchmark categories make new phenomena measurable.

Prior to the introduction of prediction benchmarks, predictive performance was difficult to evaluate systematically.

Prior to the introduction of robustness benchmarks, robustness was difficult to measure directly.

Similarly, mechanism attribution remains difficult to evaluate when it is not treated as an explicit benchmark target.

The introduction of mechanism labels transforms explanatory properties into benchmark variables.

Once mechanisms become benchmark targets, several previously difficult-to-measure quantities become directly evaluable.

Mechanism Attribution Accuracy

Can the model identify the structural explanation responsible for a trajectory?

Governing Constraint Identification

Can the model identify the constraint exerting the greatest influence over future behavior?

Outcome–Mechanism Divergence

Does outcome prediction remain coupled to mechanism attribution, or can they separate?

Structural Alignment

Does the inferred explanation correspond to the benchmark-generating mechanism?

Failure Reconstruction

Can the model reconstruct the structural pathway that produced an outcome?

These quantities are difficult to evaluate within conventional outcome-oriented benchmarks because the relevant targets are not typically represented within benchmark labels.

Mechanism labels make them measurable directly.

20.5 Structural Ambiguity as an Evaluation Resource

A recurring theme throughout this paper has been the role of structural ambiguity.

Traditional benchmark design often treats ambiguity as something to be minimized.

Constraint Geometry treats certain forms of ambiguity as essential.

This reflects a broader shift in evaluation philosophy.

The objective is not merely to determine whether a model can arrive at a correct answer.

The objective is to determine whether it can distinguish among multiple plausible explanations.

This distinction becomes meaningful only when multiple explanations remain viable.

Structural ambiguity therefore functions for explanation benchmarks in much the same way that difficult edge cases function for prediction benchmarks.

Ambiguity is not preserved because uncertainty is desirable.

It is preserved because explanation cannot be evaluated meaningfully when only a single explanation remains possible.

Structural ambiguity creates situations in which:

- Outcome recognition becomes insufficient
- Shortcut solutions become unreliable
- Mechanism attribution remains necessary

The framework consequently treats ambiguity not as noise but as signal.

More specifically, it treats ambiguity as a carrier of mechanism-attribution information.

20.6 Implications for AI Evaluation

If structural reasoning proves measurable through benchmark design, several implications follow for AI evaluation.

More Interpretable Performance

A model that predicts correctly and attributes mechanisms correctly differs from a model that predicts correctly while consistently misidentifying mechanisms.

Conventional outcome metrics often cannot distinguish between these situations.

Mechanism-attribution metrics can.

More Granular Evaluation

Performance can be decomposed into:

- Outcome recognition
- Mechanism attribution
- Dominance attribution
- Failure reconstruction
- Structural alignment

This creates a richer evaluation space than outcome prediction alone.

Capability Isolation

Structural reasoning benchmarks may also provide a mechanism for capability isolation.

By evaluating mechanism attribution separately from outcome prediction, benchmark designers can investigate whether the two targets rely on identical or partially independent forms of reasoning.

This possibility follows directly from the dual-target architecture introduced earlier.

Better Understanding of Reasoning Behavior

Rather than asking only:

Did the model reach the correct answer?

evaluation can also ask:

Did the model identify the correct explanation?

This distinction may prove useful in domains where explanation quality matters alongside predictive accuracy.

20.7 Structural Reasoning and Capability Measurement

The framework introduced in this paper does not assume that structural reasoning constitutes a distinct capability.

That remains an empirical question.

Benchmark categories often precede capability claims because measurement must exist before capability structure can be investigated.

A capability cannot be evaluated directly until benchmark environments exist in which it can be measured.

Constraint Geometry therefore occupies an intentionally modest position.

The framework does not claim to have demonstrated a new capability.

It proposes a benchmark environment in which the existence of such a capability can be investigated.

In this sense, the benchmark serves as a measurement instrument rather than a theoretical conclusion.

Its role is not to answer the question.

Its role is to make the question measurable.

20.8 Beyond Outcome Prediction

The broader significance of structural reasoning as a benchmark category lies in the possibility of extending evaluation beyond outcomes alone.

Outcome prediction remains essential.

Constraint Geometry does not challenge that.

The framework instead suggests that benchmark ecosystems may benefit from an additional layer of evaluation focused on structural explanation.

Prediction asks:

What happened?

Structural reasoning asks:

What produced what happened?

These questions are related.

They are not necessarily identical.

If the distinction proves meaningful, structural reasoning may emerge as a useful benchmark category alongside classification, prediction, retrieval, recommendation, and other established forms of evaluation.

20.9 Discussion Summary

The central argument of this paper is not that structural reasoning has already been established as a distinct capability.

The argument is narrower.

Contemporary benchmark ecosystems provide extensive evaluation of state recognition and outcome prediction.

They provide comparatively limited direct evaluation of structural explanation.

Constraint Geometry introduces a benchmark architecture designed to address that gap.

More broadly, it raises the possibility that structural reasoning may represent a useful benchmark category organized around explanation rather than prediction.

Such a category would be centered on:

- Mechanism attribution
- Structural explanation
- Governing-constraint identification
- Failure reconstruction
- Outcome–mechanism decoupling

Whether these targets ultimately reveal a distinct form of reasoning remains an empirical question.

The contribution of the framework is not to answer that question.

It is to make the question measurable.

21. Future Benchmark Layers

The Constraint Geometry Benchmark Suite introduced in this paper should be understood as an initial benchmark family rather than a complete benchmark ecosystem.

The current framework focuses primarily on mechanism attribution under conditions of structural ambiguity. It evaluates whether models can distinguish among competing explanations, identify governing constraints, and reconstruct the mechanisms responsible for observed trajectories.

These capabilities represent only one region of a broader structural reasoning landscape.

Many real-world systems exhibit forms of complexity that extend beyond the benchmark layers introduced thus far. Constraints compete, dominance relationships evolve, interventions interact, compensatory systems adapt, and multiple agents influence shared trajectories.

The benchmark layers proposed in this section represent possible extensions of the framework.

They should be understood as future research directions rather than established benchmark standards.

The purpose of these layers is not merely to increase benchmark difficulty.

The purpose is to expand the range of structural phenomena that can be evaluated directly.

Viewed collectively, these extensions suggest a broader possibility:

Constraint Geometry may ultimately develop into a family of benchmark categories organized around progressively richer forms of structural reasoning.

21.1 Constraint Competition

The benchmark family introduced in this paper largely assumes that a governing constraint can eventually be identified.

Many real systems are less cooperative.

Multiple constraints may exert influence simultaneously, with no single constraint clearly governing behavior.

In such settings, the benchmark challenge shifts from dominance attribution to competition analysis.

The central question becomes:

Which constraints are competing for control of the trajectory?

rather than:

Which constraint currently dominates?

Potential benchmark tasks include:

- Ranking active constraints by influence
- Estimating competition intensity
- Identifying emerging dominant constraints
- Predicting future dominance transitions
- Distinguishing stable competition from unstable competition

Constraint Competition benchmarks would evaluate whether models can reason about systems in which influence remains distributed rather than centralized.

These benchmarks would extend mechanism attribution from identifying dominant constraints to understanding the structure of competition among constraints.

21.2 Dominance Switching

The current framework treats dominance attribution primarily as a snapshot problem.

Future benchmark layers could evaluate dominance as a dynamic process.

Many systems experience transitions in which one governing constraint gradually yields influence to another.

Examples include:

- Repair failure transitioning into signaling failure
- Resource exhaustion transitioning into coordination failure
- Compensation decline transitioning into collapse dynamics
- Local constraints transitioning into system-wide constraints

The benchmark question becomes:

How are dominance relationships changing through time?

rather than:

Which constraint dominates at this moment?

Potential benchmark tasks include:

- Detecting dominance transitions
- Estimating transition timing
- Identifying precursor signals of dominance change
- Predicting future dominant constraints
- Reconstructing historical dominance sequences

These benchmarks would evaluate temporal mechanism attribution rather than static mechanism attribution.

They would require models to reason not only about current structure, but also about structural evolution.

21.3 Intervention Geometry

The benchmark suite introduced in this paper primarily evaluates explanation.

A natural extension is to evaluate intervention structure.

Once governing constraints have been identified, an additional question emerges:

Which intervention is structurally aligned with the governing constraint?

This differs from conventional intervention prediction.

The objective is not merely to predict whether an intervention succeeds.

The objective is to evaluate whether the intervention targets the mechanism most responsible for the trajectory.

Potential benchmark tasks include:

- Intervention ranking
- Mechanism–intervention matching
- Constraint-target alignment
- Intervention sequencing
- Intervention timing analysis

Intervention Geometry benchmarks would extend evaluation from explanation toward structurally informed action selection.

They would provide a bridge between mechanism attribution and intervention reasoning.

21.4 Adaptive Compensation

The current benchmark family treats compensation primarily as a buffering process.

Many real systems exhibit more sophisticated forms of adaptation.

Compensatory systems may learn, reorganize, redistribute resources, or alter structural relationships over time.

In such settings, the benchmark challenge becomes:

How does the system adapt to persistent constraint pressure?

Potential benchmark tasks include:

- Detecting adaptive compensation
- Identifying compensation strategies
- Distinguishing adaptation from temporary buffering
- Predicting compensation failure
- Reconstructing adaptation pathways

Adaptive Compensation benchmarks would evaluate whether models can reason about systems whose structure changes in response to stress.

21.5 Multi-Agent Constraint Systems

Many important systems are not governed by a single decision-making entity.

Instead, multiple agents influence shared trajectories simultaneously.

Examples include:

- Biological ecosystems
- Economic systems
- Organizational environments
- Distributed control systems

In these environments, constraints emerge not only from system dynamics but also from interactions among agents.

The benchmark question becomes:

How do multiple actors jointly shape structural outcomes?

Potential benchmark tasks include:

- Agent attribution
- Influence propagation
- Coalition formation
- Conflict-driven dominance shifts
- Multi-agent intervention analysis

These benchmarks would extend structural reasoning beyond single-system dynamics toward collective behavior.

21.6 Open Structural Environments

The benchmark families introduced in this paper operate under known generative rules.

A more ambitious future direction would involve partially open structural environments.

In these environments:

- Mechanisms may evolve
- Constraint relationships may change
- New structural configurations may emerge

The benchmark challenge becomes increasingly exploratory.

Rather than identifying mechanisms within a fixed environment, models would be required to reason about changing structural landscapes.

Importantly, this direction approaches the boundary between mechanism attribution and structural discovery.

As discussed in the Scope section, these remain distinct problems.

Nevertheless, open structural environments may provide a useful intermediate step between synthetic attribution tasks and broader questions of scientific reasoning.

21.7 Toward a Structural Reasoning Benchmark Ecosystem

Taken together, these benchmark layers suggest a possible long-term trajectory for the framework.

The current benchmark suite evaluates:

- Mechanism attribution
- Structural alignment
- Dominance attribution
- Failure reconstruction

Future benchmark layers could expand evaluation toward:

- Constraint competition
- Dynamic dominance
- Intervention reasoning
- Adaptive compensation
- Multi-agent structure
- Open structural environments

Viewed collectively, these extensions point toward a broader benchmark ecosystem organized around structural reasoning.

The significance of this possibility is not that structural reasoning has already been established as a distinct capability.

The significance is that benchmark evaluation may eventually be able to investigate structural reasoning across a much wider range of environments than those introduced in the present work.

The benchmark suite presented here should therefore be understood not as a final framework, but as an initial step toward a broader research program in the evaluation of structural explanation and mechanism attribution.

22. Generalization Beyond Oncology

The Constraint Geometry framework was introduced through a family of synthetic oncology-inspired benchmarks. Throughout this paper, oncology has served as the demonstration domain used to illustrate mechanism attribution, governing-constraint identification, structural ambiguity, compensation, dominance, intervention structure, and outcome–mechanism decoupling.

The choice of oncology was intentional.

Cancer-related systems naturally exhibit many of the characteristics that make mechanism attribution difficult:

- Multiple interacting processes
- Partial observability
- Competing mechanisms
- Delayed effects
- Compensation
- Dominance shifts
- Outcome–mechanism decoupling

These properties make oncology a useful environment for benchmark construction.

However, the framework itself is not specific to oncology.

Constraint Geometry is fundamentally concerned with relationships among constraints rather than any particular biological system.

The contribution of this work is therefore not an oncology benchmark.

It is a benchmark-design methodology that uses oncology as an initial demonstration environment.

The broader question addressed in this section is whether the benchmark architecture generalizes beyond the domain in which it was first introduced.

22.1 Structural Invariants Across Domains

A central premise of Constraint Geometry is that many complex systems share common structural features despite differing substantially in their physical components.

The variables change.

The structure often does not.

Systems that appear unrelated at the level of domain knowledge frequently exhibit remarkably similar patterns of interaction at the level of constraints.

Examples include:

- Competing constraints
- Compensation mechanisms
- Resource limitations
- Information degradation
- Delayed effects
- Dominance relationships
- Adaptive responses

These recurring patterns can be understood as **structural invariants**.

A structural invariant is a relationship pattern that appears across multiple domains despite differences in the underlying variables.

For example:

- Compensation appears in immune systems, power grids, supply chains, and financial markets.
- Dominance relationships appear in biological regulation, organizational systems, and industrial control environments.
- Information degradation appears in signaling pathways, communication networks, cybersecurity systems, and governance structures.

The specific variables differ.

The structural problem remains similar.

Constraint Geometry therefore focuses on structural relationships rather than domain-specific knowledge.

The benchmark objective is not to evaluate expertise in oncology.

It is to evaluate reasoning about interacting constraints.

This distinction creates the possibility of transferring benchmark principles across domains.

22.2 Why Oncology Was Chosen

A natural question follows:

Why oncology rather than another domain?

The answer is methodological rather than biomedical.

Real oncology contains many of the characteristics that make mechanism attribution difficult:

- Competing mechanisms
- Partial observability
- Delayed effects
- Adaptive responses
- Compensation
- Dominance transitions

These properties provide a rich environment for benchmark construction.

Equally important, oncology-inspired synthetic environments allow benchmark designers to specify generating mechanisms explicitly.

This creates an important advantage for benchmark evaluation.

Mechanism attribution can be assessed directly because the benchmark-generating structure is known by construction.

The significance of oncology within this framework is therefore not biomedical.

It is architectural.

Oncology serves as a source of structural complexity rather than as the subject of scientific investigation.

22.3 Infrastructure Systems

Modern infrastructure systems provide a natural environment for structural reasoning benchmarks.

Examples include:

- Electrical grids
- Water distribution networks
- Transportation systems
- Telecommunications networks

These systems contain:

- Multiple interacting constraints
- Reserve capacity
- Compensation mechanisms
- Delayed failures
- Cascading effects

A power grid may appear stable despite operating near critical capacity limits.

A transportation network may absorb disruptions through rerouting and redundancy.

Failures often emerge from interactions among constraints rather than isolated component failures.

Benchmark questions could include:

- Which constraint governs system fragility?
- Is compensation masking instability?
- Which intervention would most effectively restore resilience?

These questions closely parallel those introduced in the oncology benchmark family.

22.4 Ecological Systems

Ecological systems provide another rich environment for mechanism-attribution evaluation.

Examples include:

- Forest ecosystems
- Fisheries
- Watersheds
- Agricultural systems
- Food webs

Ecological dynamics often involve:

- Delayed responses
- Nonlinear feedback
- Compensation
- Multiple interacting pressures
- Partial observability

Population decline may result from:

- Resource depletion
- Predation shifts
- Habitat fragmentation
- Climate stress
- Disease pressure

Similar ecological outcomes may therefore arise from different governing mechanisms.

Mechanism attribution becomes essential for understanding trajectories and evaluating interventions.

22.5 Financial Systems

Financial systems exhibit many forms of structural ambiguity that align naturally with the Constraint Geometry framework.

Examples include:

- Banking systems
- Credit markets
- Investment networks
- Liquidity systems
- Systemic-risk environments

Financial instability may emerge through:

- Liquidity shortages
- Information failures
- Coordination breakdowns
- Excess leverage
- Contagion effects

Observable outcomes such as market decline, institutional failure, or volatility spikes may conceal fundamentally different underlying mechanisms.

A benchmark focused on mechanism attribution could evaluate whether models distinguish among competing structural explanations rather than merely forecasting outcomes.

22.6 Cybersecurity Systems

Cybersecurity environments are particularly well suited to mechanism-attribution benchmarks.

Security incidents often arise through interacting technical and organizational constraints.

Examples include:

- Credential compromise
- Privilege escalation
- Supply-chain attacks
- Configuration failures
- Coordination breakdowns

Identical outcomes may emerge through radically different attack pathways.

Likewise, similar attack mechanisms may produce different outcomes depending on compensatory controls and system architecture.

Benchmark tasks could evaluate:

- Attack-path attribution
- Dominance identification
- Failure reconstruction
- Intervention prioritization

These questions align naturally with the dual-target architecture introduced earlier.

22.7 Supply Chain Systems

Supply chains provide another domain characterized by interacting constraints and compensation dynamics.

Examples include:

- Manufacturing networks
- Global logistics systems
- Distribution networks
- Inventory systems

Supply chains frequently experience:

- Delays
- Bottlenecks
- Substitution effects
- Resource constraints
- Cascading disruptions

A visible shortage may arise from:

- Transportation failures
- Production constraints
- Inventory misalignment
- Information delays

The outcome may be identical.

The governing mechanism may differ substantially.

Constraint Geometry provides a framework for evaluating whether models can distinguish among these explanations.

22.8 Governance and Organizational Systems

Governance systems introduce a broader class of structural reasoning problems.

Examples include:

- Regulatory systems
- Public institutions
- Organizations
- Policy environments
- Multi-stakeholder systems

These environments often contain:

- Distributed authority
- Competing incentives
- Information asymmetries
- Adaptive behavior
- Feedback loops

Observable outcomes such as institutional failure, policy stagnation, or coordination breakdown may arise through multiple competing mechanisms.

Benchmark evaluation could therefore focus on:

- Governing constraints
- Dominance relationships
- Compensation structures
- Failure reconstruction

rather than outcome recognition alone.

22.9 Industrial Control Systems

Industrial control systems represent perhaps the closest engineering analogue to the benchmark environments introduced throughout this paper.

Examples include:

- Manufacturing plants
- Chemical processes
- Refineries
- Autonomous operational systems

These environments contain:

- Feedback control loops
- Resource constraints
- Compensation mechanisms
- Signal degradation
- Latency effects

Many failures emerge not from isolated component defects but from interactions among multiple constraints operating through time.

The benchmark question therefore becomes:

Which constraint governs the trajectory of the system?

rather than simply:

Has failure occurred?

This mirrors the central question of Constraint Geometry.

22.10 Toward a Domain-Independent Benchmark Architecture

The examples above illustrate a broader point.

The benchmark architecture introduced in this paper is not fundamentally tied to oncology.

The specific variables may change.

The structural questions remain remarkably similar.

Across domains, mechanism attribution often involves:

- Identifying governing constraints
- Evaluating compensation
- Reasoning under partial observability
- Distinguishing among competing explanations
- Reconstructing failure pathways
- Selecting structurally appropriate interventions

These recurring patterns suggest that Constraint Geometry may be better understood as a domain-independent benchmark architecture organized around structural invariants rather than domain-specific variables.

The framework therefore generalizes through structure rather than through content.

Whether this generalization ultimately proves successful remains an empirical question.

The benchmark family introduced here provides only an initial demonstration.

22.11 Oncology as a Demonstration Domain

The use of oncology throughout this paper should therefore be interpreted narrowly.

Oncology serves as:

- A benchmark construction environment
- A source of structural complexity
- A demonstration domain

It does not serve as:

- A biological theory
- A disease model
- A clinical framework
- A claim about cancer causation

The significance of oncology within this work is architectural rather than biomedical.

Its role is to provide a rich environment in which mechanism-attribution benchmarks can be constructed and evaluated.

The broader contribution is the benchmark architecture itself.

If the framework proves useful, future benchmark families may emerge across infrastructure, ecology, finance, cybersecurity, governance, supply chains, industrial control systems, and

other domains characterized by interacting constraints and structural ambiguity.

In that sense, oncology represents the starting point rather than the destination of the Constraint Geometry framework.

23. Implications

The primary contribution of this work is a benchmark architecture for evaluating mechanism attribution under controlled conditions.

The framework does not propose a new scientific theory, clinical model, or artificial intelligence architecture.

Instead, it introduces a benchmark methodology intended to make structural reasoning measurable.

If the distinction between outcome recognition and mechanism attribution proves meaningful, the implications extend beyond the specific benchmark family introduced in this paper.

The significance of Constraint Geometry therefore lies less in the oncology demonstration domain than in the broader evaluation questions it raises.

At its core, the framework asks whether benchmark ecosystems can evaluate not only what models predict, but also how they explain.

This section considers the implications of that possibility.

23.1 Implications for Benchmark Theory

The most immediate implications of this work concern benchmark design itself.

Most benchmark categories are organized around observable outputs.

Examples include:

- Classification
- Prediction
- Retrieval
- Recommendation
- Generation

Constraint Geometry proposes a different organizing principle.

Benchmark categories may also be organized around explanatory targets.

Under this view, structural explanation becomes a benchmark object in its own right.

The framework therefore raises a broader question:

Should benchmark ecosystems evaluate explanations as directly as they evaluate outcomes?

This question does not assume that explanation and prediction are distinct capabilities.

Rather, it suggests that they may represent distinct evaluation targets.

If so, benchmark architectures may need mechanisms for measuring both.

Constraint Geometry represents one attempt to operationalize that possibility.

23.2 Implications for Capability Measurement

A recurring theme throughout this paper has been the distinction between making a capability measurable and proving that it exists.

Constraint Geometry does not establish mechanism attribution as a distinct capability.

Instead, it introduces a benchmark environment in which that possibility can be investigated.

This distinction is important.

Benchmark categories often precede capability claims.

Recognition became measurable before recognition capabilities could be compared systematically.

Prediction became measurable before forecasting performance could be studied rigorously.

Similarly, structural reasoning may need benchmark environments before meaningful claims can be made regarding its nature or separability.

The framework therefore contributes a measurement instrument rather than a conclusion.

Its purpose is not to resolve questions about capability structure.

Its purpose is to make those questions empirically testable.

23.3 Implications for AI Evaluation Research

Modern AI evaluation has expanded far beyond traditional accuracy metrics.

Benchmark ecosystems now evaluate:

- Knowledge
- Reasoning
- Robustness
- Truthfulness
- Calibration
- Agentic behavior
- Tool use

Constraint Geometry proposes an additional evaluation target:

structural explanation.

More specifically, it proposes direct evaluation of:

- Mechanism attribution
- Structural alignment
- Governing-constraint identification
- Failure reconstruction
- Outcome–mechanism divergence

These quantities are difficult to evaluate when explanation remains implicit.

They become measurable when explanations themselves become benchmark targets.

The framework therefore contributes not only a benchmark suite, but also a broader perspective on what benchmark ecosystems may be capable of measuring.

23.4 Implications for Systems-Oriented Reasoning

Constraint Geometry emerged from observations that appear repeatedly across systems-oriented disciplines.

Examples include:

- Systems biology
- Complex adaptive systems
- Resilience science
- Robustness theory
- Systems engineering
- Network science

These disciplines frequently emphasize:

- Interactions rather than isolated variables
- Feedback rather than static relationships
- Compensation rather than simple causation
- Trajectories rather than snapshots

Constraint Geometry does not attempt to unify these fields.

However, it introduces a benchmark architecture that reflects many of the structural concepts they study.

The framework therefore suggests a possible bridge between:

systems-oriented theory

and

benchmark-oriented evaluation.

Rather than asking whether a systems theory is correct, the benchmark asks whether a model can reason within environments characterized by the kinds of structural relationships those theories describe.

This provides one possible path toward operationalizing systems-oriented concepts within AI evaluation.

23.5 Implications for Biomedical AI

Biomedical AI has experienced substantial advances through benchmark-driven evaluation.

Models now achieve increasingly strong performance across tasks such as:

- Diagnosis
- Prognosis
- Survival prediction
- Treatment-response prediction
- Clinical question answering

These benchmarks have significantly improved the measurement of predictive performance.

Constraint Geometry suggests a complementary evaluation target.

Many clinical decisions ultimately depend not only on predicting outcomes but also on understanding the mechanisms responsible for those outcomes.

Two patients may:

- Share a diagnosis
- Share a risk score
- Share a prognosis

yet differ substantially in the mechanisms governing their trajectories.

Likewise, identical interventions may produce different effects when applied to patients governed by different structural conditions.

Mechanism-attribution benchmarks could therefore provide a way to evaluate whether models distinguish among competing structural explanations rather than merely predicting outcomes.

Importantly, this paper does not claim that mechanism attribution is necessary for clinical competence.

Nor does it claim that success on synthetic benchmarks implies clinical understanding.

The implication is narrower:

Benchmark ecosystems may benefit from direct evaluation of structural explanation alongside prediction.

23.6 Implications for Safety-Critical Systems

Safety-critical environments often require more than outcome prediction.

Examples include:

- Aviation
- Industrial control systems
- Energy infrastructure
- Autonomous systems
- Cybersecurity environments

In these domains, understanding why a system behaves as it does may be as important as predicting what it will do.

A model that predicts failure correctly but misidentifies the governing mechanism may still support ineffective interventions.

Similarly, a system may appear stable while compensation conceals growing instability.

These situations closely resemble the benchmark conditions explored throughout this paper.

Constraint Geometry therefore suggests a benchmark category for evaluating:

- Failure reconstruction
- Governing-constraint identification
- Compensation awareness
- Dominance attribution
- Mechanism-sensitive intervention reasoning

Whether such benchmarks improve safety evaluation remains an empirical question.

The framework provides a means of investigating that question directly.

23.7 Implications for General Reasoning

The ideas introduced in this paper may extend beyond biomedical and systems-oriented domains.

Many reasoning tasks involve distinguishing among competing explanations.

Examples include:

- Scientific reasoning
- Historical analysis
- Root-cause investigation
- Engineering diagnostics
- Strategic analysis
- Organizational failure analysis

In each case, the challenge often involves more than identifying an outcome.

It involves identifying the explanation most consistent with available evidence.

Constraint Geometry raises a broader possibility:

Can mechanism attribution be evaluated as a general reasoning target?

This paper does not answer that question.

However, it provides one methodology through which it can be investigated.

If mechanism attribution proves measurable across domains, structural reasoning benchmarks may eventually emerge as a complement to existing reasoning benchmarks organized around prediction, classification, or problem solving.

23.8 A Broader Evaluation Perspective

Taken together, these implications point toward a broader view of benchmark evaluation.

Traditional benchmark architectures have been highly successful at measuring:

- Recognition
- Classification
- Prediction
- Forecasting

Constraint Geometry explores whether benchmark ecosystems can also evaluate:

- Structural explanation
- Mechanism attribution
- Governing constraints
- Failure reconstruction
- Outcome–mechanism decoupling

This does not imply a replacement of existing benchmarks.

Nor does it imply that explanation is inherently more important than prediction.

Rather, it suggests that benchmark ecosystems may eventually be capable of measuring both.

Prediction answers:

What happened?

Structural explanation asks:

What produced what happened?

The relationship between those questions remains an empirical matter.

Constraint Geometry provides a framework through which that relationship can be investigated.

23.9 Implications Summary

The broader implication of this work is methodological rather than domain-specific.

Constraint Geometry proposes that benchmark evaluation can be extended beyond observable outcomes toward the structural explanations that generate those outcomes.

If this extension proves useful, its potential relevance spans:

- Benchmark theory
- Capability measurement
- AI evaluation research
- Systems science
- Biomedical AI
- Safety-critical systems
- General reasoning

The framework makes no claim that mechanism attribution is a distinct capability, that synthetic benchmark success implies real-world understanding, or that explanation should replace prediction.

Its contribution is narrower and more practical.

It provides a benchmark architecture through which these questions can be investigated empirically.

In that sense, the ultimate implication of Constraint Geometry is not a conclusion about artificial intelligence.

It is a proposal about evaluation:

If structural reasoning matters, benchmark ecosystems should be able to measure it.

Conclusion

From State Recognition to Structural Reasoning

Benchmark-driven evaluation has become one of the primary mechanisms through which progress in artificial intelligence is measured. Across domains, benchmark performance is often interpreted as evidence of increasingly capable prediction, classification, forecasting, and reasoning.

Most contemporary benchmarks are organized around outcomes.

They evaluate whether a model can correctly identify a state, predict a future event, estimate a probability, retrieve relevant information, or select an appropriate response. These are important and valuable capabilities, and the success of modern AI systems owes much to the development of benchmark ecosystems that measure them reliably.

This paper has argued that an additional evaluation target may deserve direct attention.

Many complex systems are governed by interacting constraints, compensatory processes, delayed effects, incomplete observations, competing mechanisms, and shifting dominance relationships. In such settings, similar outcomes may arise from different structural conditions, while similar structural conditions may produce different outcomes depending on context and trajectory.

Under these conditions, outcome recognition and mechanism attribution are not necessarily identical tasks.

More fundamentally, outcomes often compress information.

A trajectory containing multiple interacting mechanisms may ultimately be represented by a single observable label:

- Failure
- Recovery
- Stability
- Progression

These labels capture what happened.

They do not necessarily preserve information about why it happened.

Mechanism attribution can therefore be understood as an attempt to recover structural distinctions that outcome labels alone may conceal.

This observation motivated the development of Constraint Geometry, a benchmark framework designed to evaluate whether models can distinguish among competing structural explanations that remain consistent with similar observations.

The framework introduced three central ideas.

First, mechanism attribution was proposed as an explicit benchmark target rather than an implicit consequence of outcome prediction.

Second, benchmark tasks were designed around structural ambiguity, ensuring that multiple explanations remain plausible and that benchmark success cannot be reduced to simple prediction, proxy detection, or shortcut learning.

Third, a dual-target architecture was introduced in which outcome recognition and mechanism attribution are evaluated separately through Outcome Accuracy and Structural Alignment Accuracy.

Together, these elements transform structural explanation from an implicit property of benchmark performance into an explicit object of evaluation.

The contribution of this work is therefore methodological rather than scientific.

Constraint Geometry is not a biological theory, a clinical framework, a disease model, or a causal-discovery system.

It is a benchmark architecture designed to investigate whether structural reasoning can be measured independently of state recognition.

The framework does not assume that mechanism attribution represents a distinct capability.

Nor does it assume that success on synthetic structural reasoning tasks implies real-world understanding.

Instead, it provides a controlled environment in which those questions can be investigated empirically.

This distinction reflects a broader principle.

Benchmark categories often precede capability claims.

Prediction became measurable before forecasting capabilities could be compared systematically.

Robustness became measurable before robustness could be studied rigorously.

Similarly, if structural reasoning is to be evaluated meaningfully, benchmark environments must first exist in which structural reasoning can be measured directly.

Constraint Geometry is proposed as one such environment.

At its core, the paper is organized around a simple distinction.

Traditional benchmarks primarily ask:

What state is the system in?

Constraint Geometry asks:

What structure governs the trajectory of the system?

The significance of this distinction is not that one question is more important than the other.

Prediction and explanation serve different purposes.

The contribution of the framework is to create a benchmark environment in which both can be evaluated explicitly.

Whether outcome recognition and mechanism attribution ultimately reveal different dimensions of reasoning remains an open empirical question.

The purpose of Constraint Geometry is not to answer that question in advance.

It is to make the question measurable.

More broadly, the framework raises the possibility that benchmark ecosystems may eventually expand beyond the evaluation of states and outcomes toward the evaluation of structural explanations themselves.

If structural reasoning matters, benchmark ecosystems should be able to measure it.

Constraint Geometry is offered as one step toward that goal.

Appendix A
Constraint Geometry Design Rules

Purpose

The Constraint Geometry framework is built on a simple principle:

A benchmark intended to evaluate mechanism attribution should require mechanism attribution for successful performance.

This principle has important implications for benchmark construction.

A benchmark may contain substantial structural complexity without actually evaluating structural reasoning.

If simpler strategies remain sufficient for success, benchmark performance may primarily reflect:

- Pattern recognition
- Proxy detection
- Threshold identification
- Shortcut learning
- Correlation extraction

rather than mechanism attribution.

The purpose of the design rules presented in this appendix is therefore not to increase benchmark difficulty.

Their purpose is to preserve mechanism-attribution signal while reducing opportunities for capability substitution.

Collectively, these rules define the benchmark-design philosophy underlying the Constraint Geometry Benchmark Suite.

A.1 Design Goal

The overarching design goal is:

Preserve mechanism-attribution signal.

A benchmark should be constructed such that successful performance requires engagement with structural explanations rather than simpler predictive shortcuts.

The benchmark should evaluate:

Why did this trajectory occur?

rather than merely:

What outcome occurred?

All subsequent design principles derive from this objective.

A.2 Necessary Condition: Structural Ambiguity

Mechanism attribution becomes meaningful only when multiple explanations remain plausible.

Constraint Geometry therefore treats structural ambiguity as a design requirement rather than a design flaw.

Definition

Structural ambiguity exists whenever multiple structural explanations remain consistent with the available evidence.

Examples include:

- Equifinality
- Compensation
- Partial observability
- Contested dominance
- Delayed effects
- Signal degradation
- Incomplete trajectories

Structural ambiguity is the condition that makes mechanism attribution necessary.

If observations uniquely determine the explanation, attribution becomes trivial.

If outcomes uniquely determine mechanisms, attribution collapses into prediction.

Structural ambiguity preserves the distinction between the two.

Structural Ambiguity and Equifinality

Structural ambiguity arises whenever multiple structural explanations remain consistent with available evidence.

Equifinality represents one important source of such ambiguity, but not the only one.

Ambiguity may also arise from compensation, partial observability, contested dominance, delayed effects, signal degradation, or incomplete trajectories.

The objective is not to maximize ambiguity.

The objective is to preserve sufficient ambiguity that mechanism attribution remains a meaningful benchmark task.

A.3 Design Axiom

Mechanism Attribution Must Be Separable from Outcome Prediction

This principle defines the central requirement of the framework.

It is not merely a benchmark rule.

It is the design axiom from which the remaining rules follow.

A benchmark cannot evaluate mechanism attribution if mechanisms can be recovered directly from outcomes.

When outcome labels uniquely determine mechanism labels:

```
Outcome
   ↓
Mechanism
```

mechanism attribution becomes redundant.

Constraint Geometry therefore requires partial independence between:

- Outcome recognition
- Mechanism attribution

This requirement preserves two forms of structural ambiguity.

Many-to-One Mapping

Different mechanisms → Same outcome

One-to-Many Mapping

Same mechanism → Different outcomes

Together these mappings create the possibility of outcome–mechanism decoupling and allow both targets to be evaluated separately.

The remaining rules can be understood as practical methods for preserving this separation.

A.4 Implementation Rules

The rules that follow operationalize the design axiom.

Their purpose is to prevent benchmark performance from collapsing into simpler strategies that bypass mechanism attribution.

Rule 1: No Single Variable Should Determine the Label

No individual variable should be sufficient to determine the correct benchmark label.

Mechanism attribution should emerge from relationships among variables rather than isolated measurements.

Failure Mode Addressed: Single-variable separation

Rule 2: Different Mechanisms Must Sometimes Produce Identical Outcomes

Multiple structural explanations should occasionally generate the same observable outcome.

Failure Mode Addressed: Equifinality collapse

Rule 3: Identical Mechanisms Must Sometimes Produce Different Outcomes

Compensation, timing, intervention, and contextual factors should occasionally alter outcomes.

Failure Mode Addressed: Hidden determinism

Rule 4: High Burden Must Sometimes Remain Stable

Compensation, reserve capacity, redundancy, or adaptation should occasionally preserve stability despite substantial stress.

Failure Mode Addressed: Proxy prediction

Rule 5: Low Burden Must Sometimes Fail

Critical constraints, signal failures, or dominance shifts should occasionally produce instability despite modest visible burden.

Failure Mode Addressed: Proxy prediction

Rule 6: Observed Stability Must Sometimes Conceal Instability

Observable stability should not always imply structural stability.

Failure Mode Addressed: Correlation without structure

Rule 7: Compensation Must Sometimes Override Dominant Signals

Compensation should occasionally alter trajectories that otherwise appear deterministic.

Failure Mode Addressed: Shortcut learning

Rule 8: Dominance Must Sometimes Be Ambiguous

Several constraints should occasionally exert comparable influence simultaneously.

These failure modes are valuable because they create situations in which simple signal-ranking strategies fail, thereby reducing opportunities for capability substitution.

Failure Mode Addressed: Hidden determinism

Rule 9: Partial Observability Must Sometimes Remain Stable

Missing information should not automatically imply instability.

Failure Mode Addressed: Proxy prediction from uncertainty

Rule 10: Outcome and Mechanism Labels Must Not Collapse

The benchmark must contain cases involving:

- Same outcome, different mechanisms
- Different outcomes, same mechanism

Failure Mode Addressed: Capability substitution

A.5 Capability Isolation

The design rules can be understood as a form of capability isolation.

Traditional benchmark construction often focuses on creating tasks that are difficult.

Constraint Geometry focuses on creating tasks that are diagnostically informative.

Difficulty alone does not establish validity.

A benchmark may be extremely difficult while still evaluating the wrong capability.

Conversely, a relatively simple benchmark may provide strong evidence that a capability is present if alternative explanations for success have been controlled.

The purpose of the design rules is therefore not to maximize difficulty.

It is to reduce alternative pathways to success.

In this sense, the rules function analogously to experimental controls.

They attempt to isolate mechanism attribution as the capability being measured.

Whether that capability ultimately proves distinct remains an empirical question.

The benchmark architecture is designed to make that question testable.

A.6 Design Logic

The logic underlying the framework can be summarized as follows:

Principle	Function
Structural Ambiguity	Creates the need for mechanism attribution
Mechanism Attribution Must Be Separable from Outcome Prediction	Preserves mechanism-attribution signal
Structural Diversity Rules	Prevent deterministic label recovery
Stability–Burden Decoupling Rules	Prevent proxy prediction
Constraint Dynamics Rules	Prevent shortcut learning
Information Constraint Rules	Prevent uncertainty proxies
Outcome–Mechanism Separation Rules	Prevent capability substitution

Together these principles increase the likelihood that benchmark success reflects engagement with structural explanations rather than simpler predictive strategies.

A.7 Design Rule Summary

Design Rule	Primary Failure Mode Addressed
No Single Variable Should Determine the Label	Single-variable separation
Different Mechanisms Must Sometimes Produce Identical Outcomes	Equifinality collapse
Identical Mechanisms Must Sometimes Produce Different Outcomes	Hidden determinism
High Burden Must Sometimes Remain Stable	Proxy prediction
Low Burden Must Sometimes Fail	Proxy prediction
Observed Stability Must Sometimes Conceal Instability	Correlation without structure
Compensation Must Sometimes Override Dominant Signals	Shortcut learning
Dominance Must Sometimes Be Ambiguous	Hidden determinism
Partial Observability Must Sometimes Remain Stable	Proxy prediction
Outcome and Mechanism Labels Must Not Collapse	Capability substitution

---

A.8 Appendix Summary

The Constraint Geometry Design Rules define the conditions under which mechanism attribution can remain a meaningful benchmark target.

Their purpose is not to guarantee structural reasoning.

Their purpose is to preserve the possibility of measuring it.

Collectively, these rules maintain:

- Structural ambiguity
- Mechanistic diversity
- Outcome–mechanism separation
- Resistance to shortcut learning
- Resistance to proxy prediction
- Resistance to capability substitution

The result is a benchmark architecture in which successful performance is more likely to reflect engagement with structural explanations rather than reliance on simpler predictive strategies.

In this sense, the design rules serve as the benchmark analogue of experimental controls.

They preserve the signal that the benchmark is intended to measure.

Appendix B
Benchmark Family Summaries

Purpose

The Constraint Geometry Benchmark Suite (CGBS) is organized as a progression of benchmark layers designed to evaluate increasingly sophisticated forms of mechanism attribution.

Each benchmark focuses on a specific structural reasoning problem while preserving the core principles of the framework:

- Structural ambiguity
- Outcome–mechanism separation
- Mechanistic diversity
- Resistance to shortcut learning
- Dual-target evaluation

Taken together, the benchmark family forms a hierarchy of mechanism-attribution challenges extending from basic instability recognition to governing-constraint identification.

The purpose of this appendix is not merely to describe individual benchmarks.

It is to show how the benchmark family evolves through progressively richer forms of structural reasoning.

B.1 Progressive Mechanism Attribution

The benchmark family is organized around a simple progression.

Early benchmark layers focus primarily on detecting the presence of structural instability.

Later benchmark layers focus increasingly on identifying the mechanisms responsible for that instability.

The progression can therefore be understood as a movement from:

State Recognition

toward

Structural Explanation

and ultimately toward

Governing-Constraint Identification

Each benchmark introduces a new source of structural ambiguity that must be resolved through mechanism attribution.

The resulting sequence gradually increases the amount of structural reasoning required for successful performance.

Viewed collectively, the benchmark family can be interpreted as a curriculum of mechanism-attribution challenges.

B.2 CGBS-v0.1: Constraint Geometry

Benchmark Question

Is instability present within the system?

Structural Focus

Constraint interaction and stability classification.

Structural Ambiguity Introduced

Instability emerges from interacting constraints rather than isolated variables.

Target Structure

The benchmark evaluates whether instability arises from relationships among constraints rather than from any single measurement.

Mechanism Targets

Examples include:

- Stable Constraint Configuration
- Emerging Constraint Instability
- Constraint Overload
- Constraint Interaction Failure

Case Types

- Stable systems
- Unstable systems
- High-burden stable systems
- Low-burden unstable systems

Anti-Shortcut Design

- No single-variable separation
- Stability–burden decoupling
- Mechanistic diversity

Scoring

Target A

Outcome Recognition

Target B

Mechanism Attribution

Metrics

- Outcome Accuracy
- Structural Alignment Accuracy (SAA)

Structural Contribution

Introduces the distinction between observable state and underlying structural condition.

B.3 CGBS-v0.2: Compensation Geometry

Benchmark Question

Is compensation masking instability?

Structural Focus

Compensation as a source of structural ambiguity.

Structural Ambiguity Introduced

Observable stability no longer guarantees structural stability.

Target Structure

Systems that remain operational despite accumulating instability.

Mechanism Targets

Examples include:

- Effective Compensation
- Compensation Exhaustion
- Compensation Failure
- Compensation Recovery

Case Types

- Stable compensated systems
- Stable uncompensated systems
- Failing compensated systems
- Compensation-collapse systems

Anti-Shortcut Design

- Stable appearance does not imply structural stability
- Compensation occasionally overrides dominant signals

Scoring

- Outcome Accuracy
- Compensation Attribution Accuracy
- Structural Alignment Accuracy

Structural Contribution

Introduces hidden instability and the first major form of outcome–mechanism decoupling.

B.4 CGBS-v0.3: Readability Collapse

Benchmark Question

Can the system still interpret critical signals?

Structural Focus

Signal interpretation and information degradation.

Structural Ambiguity Introduced

Signal absence and signal degradation become distinct explanations.

Target Structure

Failure emerging from impaired readability rather than missing information.

Mechanism Targets

Examples include:

- Healthy Readability
- Partial Readability Collapse
- Severe Readability Collapse
- Readability Recovery

Case Types

- Missing-signal cases
- Degraded-signal cases
- Ambiguous-information cases
- Readability-collapse cases

Anti-Shortcut Design

- Missing information and degraded information remain separable
- Outcome labels do not uniquely determine readability status

Scoring

- Outcome Accuracy
- Readability Attribution Accuracy
- Structural Alignment Accuracy

Structural Contribution

Introduces information quality as a mechanism-attribution target.

B.5 CGBS-v0.4: Signal Alignment

Benchmark Question

Are available signals aligned with system needs?

Structural Focus

Alignment between information and operational requirements.

Structural Ambiguity Introduced

Information may be readable yet structurally irrelevant.

Target Structure

Systems receiving information that is present but poorly aligned with what adaptation requires.

Mechanism Targets

Examples include:

- High Alignment
- Partial Misalignment
- Severe Misalignment
- Alignment Recovery

Case Types

- Aligned systems
- Misaligned systems
- Readable-but-misaligned systems
- Ambiguous alignment systems

Anti-Shortcut Design

- Signal presence does not imply usefulness
- Readability and alignment remain independent

Scoring

- Outcome Accuracy
- Alignment Attribution Accuracy
- Structural Alignment Accuracy

Structural Contribution

Separates information availability from information relevance.

B.6 CGBS-v0.5: Signal Latency Boundary

Benchmark Question

Has delay exceeded recoverable limits?

Structural Focus

Time-dependent failure and delayed adaptation.

Structural Ambiguity Introduced

Timing becomes separable from burden.

Target Structure

Systems in which timing rather than severity governs outcome.

Mechanism Targets

Examples include:

- Timely Adaptation
- Emerging Latency Risk
- Latency Boundary Breach
- Irrecoverable Delay

Case Types

- Early intervention
- Late intervention
- Delayed recovery
- Latency-driven failure

Anti-Shortcut Design

- Similar burdens produce different outcomes
- Timing rather than severity governs trajectory

Scoring

- Outcome Accuracy
- Latency Attribution Accuracy
- Structural Alignment Accuracy

Structural Contribution

Introduces temporal mechanism attribution.

B.7 CGBS-v0.6: Missing Signal Detection

Benchmark Question

Is a critical signal absent?

Structural Focus

Reasoning under incomplete information.

Structural Ambiguity Introduced

Missing information becomes structurally significant only in some cases.

Target Structure

Systems operating under partial observability.

Mechanism Targets

Examples include:

- Non-Critical Missing Signal
- Critical Missing Signal
- Recoverable Missing Signal
- Dominance-Hiding Missing Signal

Case Types

- Stable partial-observability systems
- Unstable partial-observability systems
- Hidden-dominance systems
- Signal-loss systems

Anti-Shortcut Design

- Missing information does not imply failure
- Stable systems exist under incomplete observability

Scoring

- Outcome Accuracy
- Missing-Signal Attribution Accuracy
- Structural Alignment Accuracy

Structural Contribution

Introduces uncertainty-sensitive mechanism attribution.

B.8 CGBS-v0.7: Constraint Dominance

Benchmark Question

Which constraint governs the trajectory?

Structural Focus

Dominance attribution and governing-constraint identification.

Structural Ambiguity Introduced

Multiple active constraints compete simultaneously for influence.

Target Structure

Systems containing several plausible explanations.

Mechanism Targets

Examples include:

- Repair Dominance
- Readability Dominance
- Alignment Dominance
- Compensation Dominance
- Latency Dominance

Case Types

- Clear dominance
- Balanced constraints
- Contested dominance
- Dominance switching
- Hidden dominance

Anti-Shortcut Design

- Multiple active constraints
- Ambiguous dominance relationships
- Compensation-induced misclassification

Scoring

- Outcome Accuracy
- Dominance Attribution Accuracy
- Structural Alignment Accuracy

Structural Contribution

Introduces governing-constraint identification as the highest-level mechanism-attribution task within the current benchmark family.

B.9 Benchmark Progression Architecture

The benchmark family can be understood as a progression through increasingly sophisticated sources of structural ambiguity.

Benchmark	Core Question	New Structural Challenge
v0.1 Constraint Geometry	Is instability present?	Constraint interaction
v0.2 Compensation Geometry	Is compensation masking instability?	Hidden instability
v0.3 Readability Collapse	Can the system interpret signals?	Information degradation
v0.4 Signal Alignment	Are signals aligned with need?	Information relevance
v0.5 Signal Latency Boundary	Has delay exceeded recoverable limits?	Temporal structure
v0.6 Missing Signal Detection	Is a critical signal absent?	Partial observability
v0.7 Constraint Dominance	Which constraint governs the trajectory?	Competing explanations

Viewed this way, each benchmark layer introduces a new mechanism-attribution challenge while preserving the core design principles of the framework.

B.10 Family-Level Evaluation Objectives

Across all benchmark layers, the Constraint Geometry Benchmark Suite evaluates five recurring dimensions:

1. Mechanism Attribution
2. Structural Alignment
3. Governing-Constraint Identification
4. Outcome–Mechanism Decoupling
5. Failure Reconstruction

Each benchmark layer emphasizes a different aspect of structural reasoning while remaining compatible with the dual-target architecture.

Together these dimensions define the common evaluation language of the benchmark family.

B.11 Appendix Summary

The Constraint Geometry Benchmark Suite is designed as a layered benchmark family rather than a collection of independent tasks.

Each benchmark introduces a distinct source of structural ambiguity while preserving:

- Outcome–mechanism separation
- Structural ambiguity
- Anti-shortcut controls
- Dual-target evaluation

Viewed collectively, the benchmark family forms a progression from instability recognition toward increasingly sophisticated forms of mechanism attribution and governing-constraint identification.

The benchmark suite should therefore be understood not as a collection of oncology tasks, but as a structured curriculum of structural reasoning evaluations organized around progressively richer forms of mechanism attribution.

Its purpose is to provide a controlled environment in which structural reasoning can be measured, compared, and studied empirically.

Appendix C
Scoring Architecture

Purpose

The Constraint Geometry framework introduces a benchmark category centered on mechanism attribution. As a result, conventional outcome-oriented scoring alone is insufficient.

Traditional benchmark architectures typically evaluate a single target:

Was the outcome predicted correctly?

Constraint Geometry evaluates two targets:

- Outcome Recognition
- Mechanism Attribution

The scoring architecture must therefore support both.

More fundamentally, the framework is designed to evaluate not only predictive performance, but also the relationship between prediction and explanation. The objective is not merely to determine whether a model reaches the correct answer, but whether it arrives at that answer through a structurally appropriate explanation.

This appendix describes the evaluation framework used throughout the Constraint Geometry Benchmark Suite (CGBS) and introduces the core metrics used to assess structural reasoning performance.

C.1 Design Principles

The scoring architecture is built around four principles.

Principle 1: Outcome and Mechanism Should Be Evaluated Separately

Outcome prediction and mechanism attribution represent distinct benchmark targets.

Scoring should not assume that success on one implies success on the other.

Principle 2: Structural Alignment Should Be Measurable

Mechanism attribution must be evaluated against known benchmark-generating structures.

Principle 3: Outcome–Mechanism Divergence Should Be Visible

The scoring system should detect situations in which prediction succeeds while explanation fails.

Principle 4: Benchmark Interpretation Should Remain Transparent

Scores should correspond to clearly defined benchmark targets rather than composite metrics that obscure performance sources.

Together, these principles support a broader objective:

making prediction, explanation, and their relationship independently measurable.

C.2 Dual-Target Evaluation Architecture

Every benchmark instance contains two labels.

Target A

Outcome Label

Examples:

- Stable
- Unstable
- Recovery
- Failure

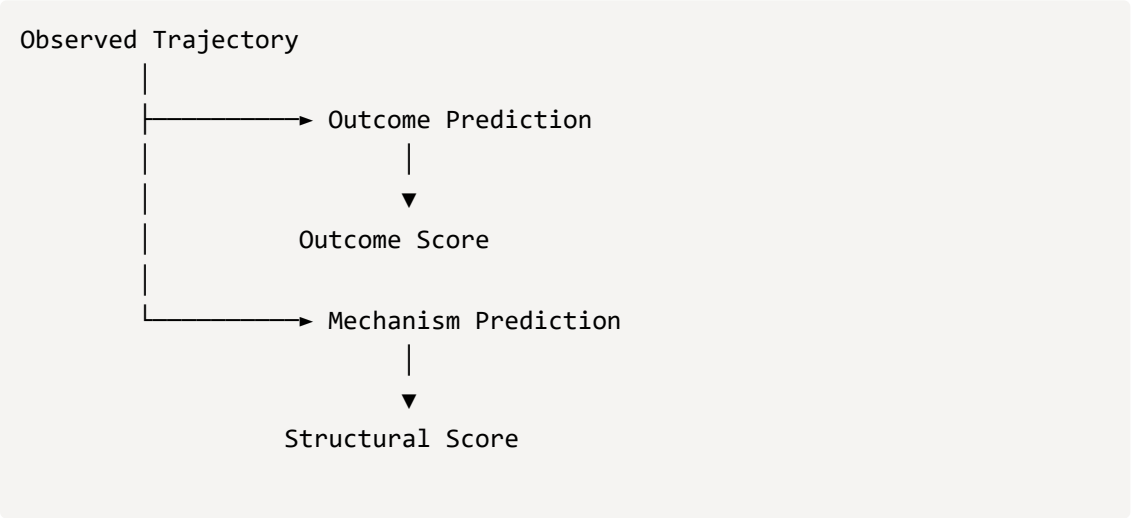
Target B

Mechanism Label

Examples:

- Repair Capacity Collapse
- Readability Collapse
- Compensation Exhaustion
- Alignment Failure
- Latency Failure
- Dominance Shift

These labels are evaluated independently.



This architecture enables outcome performance and mechanism performance to be compared directly.

The central question is not whether one target is more important than the other.

The central question is whether the two targets remain coupled under conditions of structural ambiguity.

C.3 Measurement Hierarchy

The scoring architecture can be understood as a three-level measurement framework.

Level 1: Prediction

Measures whether the model identifies the correct outcome.

Examples:

- Outcome Accuracy
- Precision
- Recall
- F1 Score

Level 2: Explanation

Measures whether the model identifies the correct structural explanation.

Examples:

- Structural Alignment Accuracy
- Mechanism Attribution Accuracy
- Governing Constraint Accuracy
- Failure Reconstruction Accuracy

Level 3: Decoupling

Measures the relationship between prediction and explanation.

Examples:

- Outcome–Mechanism Decoupling
- Attribution Error Distribution

This hierarchy mirrors the central logic of the framework:

```
Prediction
     │
     ▼
Explanation
     │
     ▼
Prediction–Explanation Relationship
```

Constraint Geometry is concerned not only with performance at each level, but also with how performance changes between levels.

C.4 Outcome Evaluation

Outcome evaluation follows conventional benchmark practice.

The objective is to determine whether the observable state associated with a benchmark case is correctly identified.

Outcome Accuracy

Outcome Accuracy is defined as:

$$\frac{\text{Correct Outcome Predictions}}{\text{Total Benchmark Cases}}$$

Outcome Accuracy measures state-recognition performance.

Examples include:

- Stable vs Unstable
- Recovery vs Failure
- High Risk vs Low Risk

Outcome Accuracy serves as the primary predictive metric throughout the benchmark family.

C.5 Mechanism Evaluation

Mechanism evaluation measures attribution performance.

The objective is to determine whether the model identifies the structural explanation responsible for generating a trajectory.

Mechanism Attribution Accuracy

Mechanism Attribution Accuracy is defined as:

```
Correct Mechanism Predictions
_____________________________

Total Benchmark Cases
```

Unlike Outcome Accuracy, this metric evaluates explanation quality rather than outcome correctness.

The benchmark question becomes:

Which mechanism generated this trajectory?

rather than:

What outcome occurred?

Mechanism Attribution Accuracy represents the most direct measure of explanation classification within the framework.

C.6 Structural Alignment Accuracy (SAA)

The primary explanatory metric used throughout the framework is:

Structural Alignment Accuracy (SAA)

SAA measures the degree of correspondence between:

- Predicted mechanism
- Generating mechanism

Conceptually:

```
Predicted Explanation
          |
```

```
          ▼
Generating Mechanism
```

A prediction receives credit when the inferred explanation aligns with the benchmark-generating mechanism.

Structural Alignment Accuracy is conceptually analogous to Outcome Accuracy, but applies to explanations rather than outcomes.

Where Outcome Accuracy measures whether the model identifies the correct state, SAA measures whether it identifies the correct structural explanation.

Interpretation

High SAA indicates:

- Accurate mechanism attribution
- Accurate structural explanation
- Successful governing-constraint identification

Low SAA indicates:

- Attribution failure
- Mechanism confusion
- Structural misalignment

SAA serves as the primary explanatory metric of the Constraint Geometry framework.

Mechanism Attribution Accuracy, Governing Constraint Accuracy, and Failure Reconstruction Accuracy can be understood as specialized forms of structural alignment applied to particular benchmark layers.

C.7 Outcome–Mechanism Decoupling (OMD)

One of the central objectives of the framework is to evaluate whether outcome prediction and mechanism attribution diverge.

To support this objective, the scoring architecture introduces:

Outcome–Mechanism Decoupling (OMD)

OMD measures the difference between:

- Outcome Accuracy
- Structural Alignment Accuracy

Conceptually:

```
OMD = Outcome Accuracy − SAA
```

Interpretation

Low OMD

Outcome prediction and mechanism attribution remain closely coupled.

High OMD

Outcome prediction substantially exceeds mechanism attribution.

This suggests successful prediction without successful explanation.

Importantly, OMD is not primarily a performance metric.

It is a benchmark diagnostic.

Its purpose is to reveal whether prediction and explanation remain coupled or diverge under specific benchmark conditions.

Large OMD values indicate that benchmark success may be driven by outcome recognition without corresponding explanatory performance.

OMD therefore provides a direct measure of outcome–mechanism divergence.

C.8 Failure Reconstruction Accuracy

Several benchmark layers evaluate the ability to reconstruct the pathway responsible for a trajectory.

Examples include:

- Compensation Collapse
- Latency Failure
- Readability Collapse
- Dominance Switching

For these tasks the framework introduces:

Failure Reconstruction Accuracy (FRA)

FRA measures whether the model correctly identifies the failure pathway responsible for the observed outcome.

The benchmark question becomes:

What produced the observed failure?

rather than:

Did failure occur?

FRA extends mechanism attribution into multi-stage structural explanations and provides a direct measure of failure reconstruction performance.

C.9 Governing Constraint Accuracy

Benchmark layers involving dominance attribution require an additional metric.

Governing Constraint Accuracy (GCA)

GCA evaluates whether the model correctly identifies the constraint exerting the greatest influence over future trajectory.

Examples include:

- Repair Dominance
- Compensation Dominance
- Readability Dominance
- Alignment Dominance

The benchmark question becomes:

Which constraint governs the system?

GCA therefore measures dominance attribution directly and serves as the primary evaluation metric for governing-constraint identification tasks.

C.10 Ambiguity-Aware Evaluation

Many benchmark cases deliberately preserve structural ambiguity.

As a result, not all attribution errors are equally informative.

The framework distinguishes conceptually between:

Adjacent Attribution Errors

The model selects a mechanism that is structurally related to the correct mechanism.

Example:

```
Repair Collapse
        ↓
Compensation Exhaustion
```

Distant Attribution Errors

The model selects a structurally unrelated explanation.

Example:

```
Repair Collapse
        ↓
Latency Failure
```

Future benchmark versions may incorporate ambiguity-aware scoring that differentiates between these error categories.

The initial benchmark family uses exact-match scoring for simplicity, transparency, and reproducibility.

C.11 Dominance Attribution Scoring

Dominance-oriented benchmark layers require additional evaluation categories.

Clear Dominance Cases

One governing mechanism exists.

Contested Dominance Cases

Multiple mechanisms exert comparable influence.

Dominance Switching Cases

Governing mechanisms change through time.

Hidden Dominance Cases

The dominant mechanism remains partially unobserved.

Performance can therefore be analyzed separately across dominance conditions.

This provides a more detailed view of mechanism-attribution behavior and helps identify specific dominance-related failure modes.

C.12 Benchmark-Level Reporting

Each benchmark layer reports three categories of performance.

Predictive Metrics

- Accuracy
- Precision
- Recall
- F1 Score

Explanatory Metrics

- Structural Alignment Accuracy
- Mechanism Attribution Accuracy
- Governing Constraint Accuracy (when applicable)
- Failure Reconstruction Accuracy (when applicable)

Decoupling Metrics

- Outcome–Mechanism Decoupling
- Attribution Error Distribution

This reporting structure preserves visibility into predictive performance, explanatory performance, and the relationship between the two.

C.13 Interpretation Matrix

Dual-target evaluation creates four primary benchmark outcomes.

Outcome Accuracy	SAA	Interpretation
High	High	Outcome and mechanism identified correctly
High	Low	Prediction without explanation
Low	High	Structural understanding despite outcome uncertainty

Low	Low	Neither outcome nor mechanism identified

Traditional benchmark architectures typically distinguish only between correct and incorrect outcomes.

Constraint Geometry introduces an additional explanatory dimension, making it possible to evaluate whether predictions are supported by structurally appropriate explanations.

C.14 Why Dual-Target Scoring Matters

The scoring architecture reflects the central hypothesis of the framework.

If outcome prediction and mechanism attribution are effectively identical tasks:

- Outcome Accuracy and SAA should remain tightly coupled.
- OMD should remain low.

If meaningful separation exists:

- Outcome Accuracy may remain high.
- SAA may decline.
- OMD may increase.

The purpose of dual-target scoring is not merely to measure performance.

It is to determine whether prediction and explanation remain coupled under conditions of structural ambiguity.

The scoring framework therefore transforms a theoretical question into a measurable one.

C.15 Scoring Architecture Summary

The Constraint Geometry scoring architecture evaluates three distinct dimensions.

Predictive Performance

Measured through Outcome Accuracy and conventional classification metrics.

Explanatory Performance

Measured through Structural Alignment Accuracy and mechanism-attribution metrics.

Prediction–Explanation Relationship

Measured through Outcome–Mechanism Decoupling and related diagnostic measures.

Together these dimensions support the central objective of the framework:

evaluating whether models can identify not only what happened, but also the structural explanation responsible for why it happened.

The scoring architecture therefore serves as the measurement layer of the Constraint Geometry Benchmark Suite.

More broadly, it provides the empirical foundation for investigating whether mechanism attribution represents a benchmark target that can be measured independently of outcome

recognition.

Whether such separation ultimately exists remains an empirical question.

The purpose of the scoring architecture is not to answer that question in advance.

It is to make the question measurable.

Appendix D
Mechanism Attribution Framework

Purpose

The central objective of the Constraint Geometry framework is to evaluate mechanism attribution.

Throughout this paper, mechanism attribution has been treated as a benchmark target distinct from outcome recognition. The purpose of this appendix is to define that concept more precisely and provide a framework for understanding how mechanism attribution operates within Constraint Geometry benchmark environments.

The framework introduced here is not intended as a theory of explanation, a philosophy of causation, or a model of scientific reasoning.

Its purpose is narrower.

It defines mechanism attribution as an evaluable benchmark task.

More specifically, it provides the conceptual foundation for treating structural explanation as an explicit object of benchmark evaluation.

D.1 What Is Mechanism Attribution?

Mechanism attribution is the process of identifying the structural explanation most consistent with an observed trajectory.

Within the framework, a mechanism is not defined as a single variable, event, or observation.

Instead, a mechanism refers to a structured pattern of interacting constraints that generates system behavior.

Examples include:

- Compensation Exhaustion
- Readability Collapse
- Alignment Failure
- Latency Failure
- Dominance Shift
- Repair Capacity Collapse

These mechanisms describe structural conditions rather than observable outcomes.

Mechanism attribution therefore asks:

What structural explanation best accounts for the observed trajectory?

rather than:

What outcome occurred?

This distinction forms the foundation of the benchmark architecture.

Mechanism attribution can therefore be understood as the explanatory analogue of outcome prediction.

Outcome prediction seeks to identify what happened.

Mechanism attribution seeks to identify why it happened.

D.2 Mechanisms Versus Outcomes

A central premise of the framework is that outcomes and mechanisms are not equivalent.

Outcomes describe what happened.

Mechanisms describe what produced what happened.

Examples:

Outcome	Possible Mechanisms
Failure	Repair Collapse
Failure	Readability Collapse
Failure	Compensation Exhaustion
Failure	Latency Failure
Failure	Dominance Shift

The same outcome may therefore correspond to multiple mechanisms.

Likewise:

Mechanism	Possible Outcomes
Compensation Exhaustion	Failure
Compensation Exhaustion	Recovery
Compensation Exhaustion	Stabilization

The same mechanism may correspond to multiple outcomes.

Mechanism attribution exists because these mappings are not necessarily one-to-one.

This asymmetry creates the possibility of outcome–mechanism decoupling and provides the conceptual basis for dual-target evaluation.

D.3 Outcome Compression and Explanation Recovery

Outcomes compress information.

A trajectory containing multiple interacting constraints, compensatory processes, timing effects, and dominance relationships may ultimately be represented by a single outcome label such as:

- Failure
- Recovery
- Stability
- Progression

These labels summarize what happened.

They do not necessarily preserve information about how it happened.

Mechanism attribution can therefore be understood as a process of explanation recovery.

While outcome recognition identifies the endpoint of a trajectory, mechanism attribution attempts to reconstruct the structural conditions that generated it.

Under this view:

```
Trajectory
      ↓
Outcome Compression
      ↓
Outcome Label
```

Mechanism attribution attempts to recover information lost during that compression:

```
Outcome Label
      ↓
Explanation Recovery
      ↓
Structural Explanation
```

This distinction provides much of the motivation for the Constraint Geometry framework.

If outcomes compress structural information, benchmark evaluation may benefit from measuring whether models can recover it.

D.4 Structural Explanation

Constraint Geometry evaluates structural explanations rather than surface observations.

A structural explanation specifies:

- Active constraints
- Constraint interactions
- Dominance relationships
- Compensation processes
- Information conditions
- Trajectory dynamics

For example:

Surface Description

Failure occurred.

Structural Explanation

Failure occurred because repair demand exceeded repair capacity while compensatory reserves became exhausted.

The second explanation contains information absent from the first.

Mechanism attribution evaluates whether a model can recover that information.

Structural explanations therefore preserve distinctions that outcome labels alone may conceal.

D.5 The Attribution Problem

Mechanism attribution exists because multiple explanations may remain consistent with the same observations.

Consider:

```
Observed Outcome:
Failure
```

Possible explanations:

```
Explanation A:
Repair Collapse

Explanation B:
Readability Collapse

Explanation C:
Latency Failure

Explanation D:
Compensation Exhaustion
```

Outcome recognition alone cannot distinguish among these possibilities.

The benchmark challenge therefore becomes:

Which explanation is most consistent with the evidence?

This is the attribution problem.

Constraint Geometry is designed to evaluate performance on this problem directly.

D.6 Attribution Under Structural Ambiguity

Mechanism attribution becomes meaningful only when structural ambiguity exists.

If every observation uniquely identifies a mechanism:

```
Observation
      ↓
Mechanism
```

attribution becomes trivial.

Likewise, if outcomes uniquely identify mechanisms:

```
Outcome
      ↓
Mechanism
```

attribution collapses into prediction.

Constraint Geometry deliberately preserves ambiguity so that multiple explanations remain plausible.

This creates situations in which:

- Outcome prediction succeeds.
- Mechanism attribution remains uncertain.

Such cases provide the benchmark signal necessary for evaluating structural reasoning.

Structural ambiguity is therefore not a flaw within the framework.

It is the condition that makes mechanism attribution measurable.

D.7 Levels of Mechanism Attribution

Mechanism attribution can occur at multiple levels of specificity.

The framework distinguishes four levels.

Level 1: Outcome Attribution

Question:

What happened?

Examples:

- Stable
- Unstable
- Recovery
- Failure

This level corresponds to traditional benchmark evaluation.

Level 2: Mechanism Category Attribution

Question:

What type of mechanism produced the outcome?

Examples:

- Compensation Failure
- Signal Failure
- Constraint Failure
- Timing Failure

This level introduces structural explanation.

Level 3: Governing Constraint Attribution

Question:

Which constraint governed the trajectory?

Examples:

- Repair Dominance
- Readability Dominance
- Alignment Dominance
- Compensation Dominance

This level introduces dominance reasoning.

Level 4: Failure Pathway Attribution

Question:

How did the system arrive at the outcome?

Examples:

```
Signal Degradation
        ↓
Readability Collapse
        ↓
Misaligned Action
        ↓
Failure
```

This level introduces failure reconstruction.

D.7.1 Attribution Hierarchy

The four levels can be understood as a hierarchy of explanatory specificity.

Level	Question	Target
1	What happened?	Outcome
2	What type of mechanism produced it?	Mechanism Category
3	Which constraint governed it?	Dominance
4	How did it happen?	Failure Pathway

As attribution moves upward through this hierarchy, explanations become increasingly structural, contextual, and trajectory-dependent.

This progression mirrors the broader evolution of the Constraint Geometry benchmark family.

D.8 Mechanism Attribution and Structural Alignment

Mechanism attribution is evaluated through structural alignment.

A prediction is structurally aligned when the inferred explanation corresponds to the mechanism responsible for generating the trajectory.

Conceptually:

```
Observed Trajectory
        ↓
Predicted Explanation
        ↓
Generating Mechanism
```

Alignment occurs when:

```
Predicted Explanation
          =
Generating Mechanism
```

Structural Alignment Accuracy (SAA) measures the frequency with which this condition is satisfied.

Structural alignment is conceptually analogous to predictive accuracy, but applies to explanations rather than outcomes.

Where Outcome Accuracy measures whether a model identifies the correct state, Structural Alignment Accuracy measures whether it identifies the correct structural explanation.

This distinction provides the measurement foundation for mechanism attribution throughout the framework.

D.9 Attribution Failure Modes

Several distinct attribution failures may occur.

False Dominance Attribution

The model identifies a visible constraint rather than the governing constraint.

Compensation Blindness

The model mistakes compensated stability for genuine stability.

Readability Confusion

The model confuses degraded information with missing information.

Latency Misattribution

The model attributes failure to burden rather than timing.

Equifinality Collapse

The model treats distinct mechanisms as equivalent because they produce similar outcomes.

Pathway Reconstruction Failure

The model identifies the outcome but cannot reconstruct the structural pathway responsible for it.

These failures represent explanation errors rather than prediction errors.

Traditional benchmark metrics often fail to distinguish between them.

Constraint Geometry treats them as distinct forms of attribution failure.

D.10 Mechanism Attribution and Capability Isolation

Traditional benchmark architectures often confound multiple forms of reasoning within a single outcome metric.

Constraint Geometry seeks to isolate mechanism attribution as an explicit evaluation target.

This does not establish mechanism attribution as a distinct capability.

Rather, it creates benchmark conditions under which that possibility can be investigated empirically.

The framework therefore functions as a form of capability isolation.

Its purpose is not to prove that mechanism attribution exists as a separate competence.

Its purpose is to create a benchmark environment in which that question can be studied directly.

D.11 Mechanism Attribution Versus Causal Discovery

Mechanism attribution should not be confused with causal discovery.

The two tasks address different questions.

Task	Central Question
Outcome Prediction	What will happen?
Mechanism Attribution	What explains what happened?
Causal Discovery	What generative structure exists?

Causal discovery seeks to recover unknown generative structure from data.

Mechanism attribution evaluates attribution among competing structural explanations within known benchmark environments.

Constraint Geometry focuses on the second category.

The framework evaluates explanation selection rather than open-ended structure discovery.

D.12 Mechanism Attribution Versus Scientific Explanation

The framework also differs from scientific explanation.

Scientific explanation seeks to establish causal truth about real systems.

Constraint Geometry evaluates performance within synthetic benchmark environments.

The benchmark asks:

Can the model identify the explanation that generated this benchmark trajectory?

It does not ask:

Is this explanation scientifically true in the real world?

This distinction is critical to the scope of the framework and reflects the difference between benchmark evaluation and scientific discovery.

D.13 Why Mechanism Attribution Matters

The motivation for mechanism attribution arises from a simple observation:

Interventions operate on mechanisms rather than outcomes.

Outcomes describe what happened.

Mechanisms describe what produced what happened.

If different mechanisms produce similar outcomes, intervention decisions may depend on identifying the correct explanation rather than merely recognizing the outcome.

Mechanism attribution therefore provides information unavailable from outcome recognition alone.

Constraint Geometry evaluates whether models can recover that information.

This observation provides the practical motivation for treating mechanism attribution as an explicit benchmark target.

D.14 Framework Summary

The Mechanism Attribution Framework provides the conceptual foundation for Constraint Geometry.

The framework rests on five principles.

Principle 1

Outcomes compress information.

Principle 2

Mechanisms preserve structural information about how outcomes arise.

Principle 3

Mechanism attribution becomes meaningful only under structural ambiguity.

Principle 4

Structural alignment provides a direct method for evaluating explanations.

Principle 5

Mechanism attribution can therefore be treated as an explicit benchmark target.

Taken together, these principles define mechanism attribution as a benchmark target distinct from outcome recognition.

The purpose of Constraint Geometry is not to assume that mechanism attribution represents a distinct capability.

Its purpose is to provide a benchmark architecture in which that possibility can be measured empirically.

In this sense, mechanism attribution functions as the conceptual core of the entire framework.

It is the explanatory target that the benchmark architecture is designed to investigate.

Appendix E
Comparative Benchmark Analysis

Purpose

The Constraint Geometry framework was developed in response to a specific benchmark question:

Can mechanism attribution be evaluated directly rather than inferred from outcome prediction?

Contemporary benchmark ecosystems provide extensive evaluation of:

- Classification
- Prediction
- Forecasting
- Retrieval
- Risk estimation
- Outcome recognition

They provide comparatively limited direct evaluation of:

- Mechanism attribution
- Governing-constraint identification
- Failure reconstruction
- Outcome–mechanism decoupling
- Structural explanation

The purpose of this appendix is to clarify where Constraint Geometry sits within the broader benchmark landscape.

These comparisons are intended as conceptual analyses rather than exhaustive surveys of all benchmark families. Their purpose is not to argue that existing benchmarks are inadequate. Rather, they are intended to illustrate how a mechanism-attribution benchmark differs from benchmark architectures organized primarily around outcome recognition.

E.1 Comparative Framework

The comparisons presented throughout this appendix operate at four levels.

Level	Comparison Focus
Evaluation Target	What is being measured?
Information Structure	What labels and signals are available?
Capability Measurement	What forms of reasoning are evaluated?
Benchmark Philosophy	What assumptions guide benchmark construction?

The subsequent tables can be understood as increasingly detailed views of these four distinctions.

Together they describe the conceptual position occupied by Constraint Geometry within the broader benchmark ecosystem.

E.2 Benchmark Target Comparison

Benchmark Category	Primary Question	Primary Target
Classification	What category applies?	State
Diagnosis	What condition is present?	State
Risk Prediction	What outcome is likely?	Probability
Prognosis	What outcome will occur?	Outcome
Survival Prediction	How long until outcome?	Outcome
Treatment Response	Will intervention succeed?	Outcome
Retrieval	What information is relevant?	Information
Constraint Geometry	What structure governs the trajectory?	Structural Explanation

The central distinction is therefore not domain-specific.

It concerns what is treated as the primary object of evaluation.

E.3 Outcome Recognition Versus Structural Explanation

Evaluation Target	Outcome-Oriented Benchmarks	Constraint Geometr
Outcome Recognition	Primary Target	Target A
Mechanism Attribution	Typically Implicit	Target B
Structural Explanation	Rarely Evaluated Directly	Explicit Target
Failure Reconstruction	Rarely Evaluated Directly	Explicit Target
Governing Constraint Identification	Rarely Evaluated Directly	Explicit Target
Explanation Recovery	Rarely Evaluated	Explicit Target
Outcome–Mechanism Divergence	Not Measured	Explicitly Measured

Constraint Geometry does not remove outcome recognition.

Instead, it supplements outcome recognition with explicit evaluation of structural explanation.

E.4 Benchmark Information Structure

Property	Conventional Benchmarks	Constraint Geometry
Outcome Labels	Yes	Yes
Mechanism Labels	Rare	Yes
Generative Structure Labels	Rare	Yes
Dominance Labels	Rare	Yes
Failure Pathway Labels	Rare	Yes
Structural Ambiguity	Often Minimized	Deliberately Preserved

Traditional benchmark architectures often emphasize label clarity and separability.

Constraint Geometry deliberately preserves ambiguity when ambiguity is necessary for mechanism attribution.

E.5 What Benchmark Success Means

Benchmark Type	Success Demonstrates
Classification	Correct category recognition
Diagnosis	Correct condition identification
Risk Prediction	Accurate probability estimation
Prognosis	Accurate outcome forecasting
Survival Prediction	Accurate time-to-event estimation
Treatment Response	Accurate response prediction
Constraint Geometry	Correct structural explanation attribution

A central claim of this paper is that benchmark interpretation changes when structural explanations become explicit evaluation targets.

Success no longer means only predicting correctly.

It also means explaining correctly.

E.6 Relationship Between Outcome and Mechanism

Scenario	Outcome Recognition	Mechanism Attribution
Same Outcome, Same Mechanism	Coupled	Coupled
Same Outcome, Different Mechanisms	Coupled	Divergent
Different Outcomes, Same Mechanism	Divergent	Coupled
Compensation Masks Instability	Potentially Misleading	Informative
Partial Observability	Often Ambiguous	Structurally Informative

These cases illustrate why outcome prediction and mechanism attribution are not necessarily interchangeable.

The relationship between them becomes an empirical question rather than an assumption.

E.7 Outcome Compression and Explanation Recovery

Outcome labels compress trajectories.

A complex trajectory containing multiple constraints, compensatory processes, timing effects, and dominance relationships may ultimately be represented by a single outcome label.

Representation	Information Preserved
Outcome Label	Observable State
Mechanism Label	Structural Explanation
Failure Pathway	Trajectory History
Generative Structure	Constraint Configuration

Outcome recognition identifies the compressed representation.

Mechanism attribution seeks to recover structural information that may be lost during that compression.

This distinction lies at the center of the Constraint Geometry framework.

E.8 Structural Questions Across Benchmark Types

Benchmark Type	Typical Question
Classification	What category applies?
Diagnosis	What condition is present?
Risk Prediction	What outcome is likely?
Prognosis	What outcome will occur?
Survival Prediction	When will the outcome occur?
Treatment Response	Will treatment succeed?
Constraint Geometry	Which mechanism best explains the trajectory?

The benchmark questions themselves reveal the shift in evaluation target.

Constraint Geometry is organized around explanation rather than prediction alone.

E.9 Capability Comparison

Capability Layer	Conventional Benchmarks	Constraint Geometry
Pattern Recognition	Primary	Required
Correlation Detection	Primary	Required
Outcome Recognition	Primary	Target A

Structural Explanation	Rare	Target B
Failure Reconstruction	Rare	Explicit
Governing Constraint Identification	Rare	Explicit
Structural Alignment	Rare	Explicit
Outcome–Mechanism Decoupling Analysis	Not Measured	Explicit

Constraint Geometry therefore extends rather than replaces conventional benchmark capabilities.

Pattern recognition remains necessary.

Structural explanation becomes additionally measurable.

E.10 Shortcut Resistance Comparison

Failure Mode	Conventional Benchmarks	Constraint Geometry Response
Label Leakage	Possible	Controlled
Single Variable Separation	Common Risk	Explicitly Prevented
Proxy Prediction	Common Risk	Explicitly Prevented
Shortcut Learning	Common Risk	Reduced Through Structural Ambiguity
Hidden Determinism	Often Unexamined	Explicit Design Target
Equifinality Collapse	Common	Explicitly Prevented
Capability Substitution	Difficult to Detect	Central Evaluation Concern

Many of the design principles introduced throughout this paper exist specifically to reduce these benchmark failure modes.

The goal is not difficulty.

The goal is capability isolation.

E.11 Evaluation Metrics Comparison

Metric	Conventional Benchmarks	Constraint Geometry
Accuracy	Yes	Yes
Precision	Yes	Yes
Recall	Yes	Yes

F1 Score	Yes	Yes
Outcome Accuracy	Yes	Yes
Structural Alignment Accuracy (SAA)	No	Yes
Mechanism Attribution Accuracy	No	Yes
Governing Constraint Accuracy	No	Yes
Failure Reconstruction Accuracy	No	Yes
Outcome–Mechanism Decoupling (OMD)	No	Yes

The introduction of SAA and OMD reflects the framework's central objective:

making explanation measurable alongside prediction.

E.12 Benchmark Philosophy Comparison

Dimension	Conventional Evaluation	Constraint Geometry
Central Objective	Predict Outcomes	Attribute Mechanisms
Ambiguity	Minimize	Preserve
Explanation	Secondary	Primary Target
Outcome Recognition	Core Target	Target A
Mechanism Attribution	Often Implicit	Target B
Outcome–Mechanism Separation	Usually Assumed	Explicitly Measured
Benchmark Difficulty	Often Emphasized	Secondary
Capability Isolation	Limited	Explicit Design Goal

This comparison reflects a broader philosophical difference.

Traditional benchmarks often focus on outcome separability.

Constraint Geometry focuses on preserving mechanism-attribution signal.

E.13 State Recognition Versus Structural Reasoning

Dimension	State Recognition	Structural Reasoning
Core Question	What happened?	Why did it happen?
Primary Object	Outcome	Structural Explanation

Information Source	Observations	Relationships Among Constraints
Information Objective	Outcome Identification	Explanation Recovery
Success Criterion	Correct Outcome	Correct Mechanism
Failure Analysis	Outcome Error	Attribution Error
Intervention Relevance	Indirect	Direct
Benchmark Target	State	Mechanism

This comparison captures the central conceptual distinction developed throughout the paper.

State recognition identifies outcomes.

Structural reasoning identifies explanations.

E.14 Constraint Geometry Summary Table

Category	Constraint Geometry Contribution
Benchmark Architecture	Dual-target evaluation
Benchmark Design	Structural ambiguity preservation
Evaluation Methodology	Mechanism attribution measurement
Scoring Framework	Structural Alignment Accuracy
Failure Analysis	Outcome–mechanism decoupling
Structural Reasoning	Governing-constraint identification
Capability Isolation	Explicit evaluation target separation
Future Extensions	Competition, dominance switching, rescue geometry, closed-loop stabilization

Together these contributions define the framework's position within the broader benchmark landscape.

E.15 Appendix Summary

The comparisons presented in this appendix summarize the comparative position of Constraint Geometry within contemporary benchmark ecosystems.

The framework differs from conventional benchmark families not because it abandons outcome prediction, but because it supplements outcome prediction with explicit evaluation of structural explanation.

The central progression can be summarized as follows:

Benchmark Orientation	Primary Question
State Recognition	What state is the system in?
Outcome Prediction	What will happen?
Constraint Geometry	What structural explanation best accounts for the trajectory?

Constraint Geometry does not propose replacing existing benchmark categories.

It proposes extending benchmark evaluation toward an additional target:

mechanism attribution.

Whether mechanism attribution ultimately represents a benchmark target distinct from outcome recognition remains an empirical question.

The purpose of the framework is not to answer that question in advance.

It is to make the question measurable.

Appendix F
Future Benchmark Roadmap

Purpose

The Constraint Geometry Benchmark Suite (CGBS) introduced in this paper should be understood as an initial benchmark family rather than a completed benchmark ecosystem.

The benchmark layers presented throughout the manuscript focus primarily on mechanism attribution under conditions of structural ambiguity. They evaluate whether models can distinguish among competing explanations, identify governing constraints, and reconstruct the mechanisms responsible for observed trajectories.

These capabilities represent only one region of a broader structural reasoning landscape.

Many real-world systems involve additional forms of complexity, including:

- Competing constraints
- Dynamic dominance relationships
- Adaptive compensation
- Multi-stage interventions
- Closed-loop control
- Long-horizon adaptation
- Multi-agent coordination

The purpose of this appendix is to outline a possible roadmap for extending the framework beyond mechanism attribution.

The roadmap should be understood as exploratory rather than prescriptive.

Its purpose is not to define future benchmark standards in advance.

Its purpose is to identify natural directions along which structural reasoning evaluation may evolve.

F.1 Structural Reasoning Progression

The benchmark family introduced in this paper focuses primarily on attribution.

Future benchmark layers progressively expand the scope of evaluation toward intervention, control, adaptation, and coordination.

This progression can be summarized as follows:

Stage	Central Question
Attribution	Why did this happen?
Interaction	How do mechanisms combine?
Competition	What mechanisms are competing?
Intervention	What could change the trajectory?
Control	Can the trajectory be guided?
Adaptation	Can stability be maintained through change?
Coordination	Can multiple interacting systems be managed simultaneously?

The roadmap therefore represents a progression from explanation toward active management of trajectories.

Each stage introduces additional structural complexity while preserving the central focus of the framework:

reasoning about structure rather than outcomes alone.

F.2 Mechanism Interaction Benchmarks

Motivation

The benchmark family introduced in this paper focuses primarily on identifying governing mechanisms.

Many systems, however, are shaped by interactions among multiple mechanisms operating simultaneously.

Before reasoning about competition, a model may need to understand how mechanisms combine.

Core Question

How do multiple mechanisms interact to produce a trajectory?

Benchmark Targets

- Interaction strength
- Synergistic effects
- Antagonistic effects
- Emergent behavior
- Interaction pathways

Example Tasks

- Identify interacting mechanisms.
- Estimate interaction magnitude.
- Distinguish additive from synergistic effects.
- Predict interaction-driven instability.

Structural Contribution

Introduces mechanism composition as a benchmark target.

The focus shifts from identifying mechanisms individually to understanding how mechanisms operate together.

F.3 Constraint Competition Benchmarks

Motivation

The current benchmark family generally assumes that a dominant mechanism can eventually be identified.

Many real-world systems are less cooperative.

Multiple constraints may exert influence simultaneously without any single constraint fully governing behavior.

Core Question

Which constraints are competing to govern the trajectory?

Benchmark Targets

- Competition intensity
- Relative influence
- Emerging dominance
- Competitive stability

Example Tasks

- Rank active constraints.
- Estimate influence margins.
- Predict future dominance transitions.
- Identify unstable competition regimes.

Structural Contribution

Introduces distributed influence rather than single-mechanism explanations.

Mechanism attribution expands into competition analysis.

F.4 Dominance Switching Benchmarks

Motivation

Dominance relationships are often dynamic rather than static.

A system may transition from one governing mechanism to another as conditions evolve.

Core Question

When does governing influence shift from one constraint to another?

Benchmark Targets

- Dominance transition detection
- Transition timing
- Transition pathway reconstruction
- Future dominance prediction

Example Tasks

```
Repair Dominance
        ↓
Compensation Dominance
        ↓
Latency Dominance
```

Structural Contribution

Introduces temporal mechanism attribution.

The benchmark objective shifts from identifying dominance to tracking its evolution through time.

F.5 Rescue Geometry Benchmarks

Motivation

Many systems approach failure without becoming irrecoverable.

The critical challenge is determining whether meaningful intervention remains possible.

Core Question

Is recovery still achievable?

Benchmark Targets

- Recovery potential
- Rescue-window detection
- Recovery-path identification
- Intervention leverage estimation

Example Tasks

- Identify recoverable trajectories.
- Distinguish reversible from irreversible states.
- Estimate rescue-window width.
- Rank alternative recovery pathways.

Structural Contribution

Introduces recoverability as a benchmark target.

Rescue Geometry shifts evaluation from explaining failure to evaluating whether failure can still be prevented.

It therefore forms a natural bridge between mechanism attribution and intervention reasoning.

F.6 Intervention Sequencing Benchmarks

Motivation

Many interventions depend not only on what action is taken, but also on when it is taken and in what order.

Core Question

Which intervention sequence produces the most favorable trajectory?

Benchmark Targets

- Intervention ordering
- Sequence alignment
- Path dependency
- Timing sensitivity

Example Tasks

```
Intervention A → Intervention B
```

versus

```
Intervention B → Intervention A
```

with different outcomes.

Structural Contribution

Introduces intervention pathways as benchmark targets.

The focus shifts from identifying mechanisms to reasoning about how mechanisms can be altered.

F.7 Closed-Loop Stabilization Benchmarks

Motivation

The benchmark family introduced in this paper evaluates explanation.

Many real-world systems require control.

Control involves continuous adjustment based on feedback and changing conditions.

Core Question

Can the trajectory be stabilized through iterative intervention?

Benchmark Targets

- Feedback interpretation
- Control-sequence selection
- Adaptation latency
- Stabilization success

Example Tasks

- Multi-step intervention loops.
- Dynamic trajectory correction.
- Feedback-guided stabilization.
- Long-horizon control planning.

Structural Contribution

Extends benchmark evaluation from explanation toward control.

Previous benchmark layers evaluate whether models understand trajectories.

Closed-Loop Stabilization evaluates whether that understanding can be translated into effective control decisions over time.

F.8 Adaptive Compensation Benchmarks

Motivation

Compensation systems are often dynamic.

They learn, reorganize, adapt, and shift strategies in response to changing conditions.

The current benchmark family treats compensation primarily as a structural condition.

Future benchmark layers may treat compensation as an adaptive process.

Core Question

How does compensation evolve in response to changing constraints?

Benchmark Targets

- Compensation adaptation
- Compensation exhaustion
- Compensation reorganization
- Compensation recovery

Example Tasks

- Track compensation trajectories.
- Predict compensation collapse.
- Identify adaptive compensation pathways.
- Evaluate resilience under changing conditions.

Structural Contribution

Introduces adaptive resilience as an explicit benchmark target.

The emphasis shifts from static compensation states toward dynamic compensation behavior.

F.9 Multi-Agent Constraint Systems

Motivation

Most benchmark layers introduced in this paper evaluate single-system trajectories.

Many real-world environments involve multiple interacting systems and agents.

Examples include:

- Markets
- Institutions
- Ecosystems
- Supply chains
- Security environments

Core Question

How do interacting agents alter structural trajectories?

Benchmark Targets

- Cross-agent constraints
- Coordination dynamics
- Competition dynamics
- Shared-resource pressures

Example Tasks

- Cooperative stabilization.
- Competitive destabilization.
- Resource-allocation conflicts.
- Coordination-failure analysis.

Structural Contribution

Introduces relational constraint geometry across multiple interacting actors.

The benchmark focus expands from individual systems to networks of interacting systems.

F.10 Cross-Domain Benchmark Families

The oncology-inspired benchmark suite represents only one possible implementation of the framework.

Future benchmark families could emerge across a wide range of domains.

Domain	Example Benchmark Family
Infrastructure	Grid Stability Geometry
Ecology	Ecosystem Constraint Geometry
Finance	Systemic Risk Geometry
Cybersecurity	Attack Path Geometry
Supply Chains	Logistics Constraint Geometry
Governance	Institutional Stability Geometry
Industrial Control	Control Failure Geometry

The benchmark architecture remains unchanged.

Only the domain-specific variables differ.

The structural reasoning targets remain fundamentally the same.

F.11 Why Benchmark Complexity Increases

The future benchmark layers are not intended to be more difficult for their own sake.

Difficulty alone does not establish benchmark value.

Each new layer introduces a structural phenomenon that cannot be evaluated adequately within earlier benchmark families.

The progression therefore reflects increasing structural scope rather than increasing benchmark difficulty.

Examples include:

- Competition
- Intervention
- Adaptation
- Control
- Coordination

The objective is capability isolation rather than complexity escalation.

Each benchmark layer is intended to make a new form of structural reasoning measurable.

F.12 Structural Reasoning Maturity Model

The roadmap can be viewed as a progression of benchmark maturity.

Level	Primary Question
Level 1	Is instability present?
Level 2	What mechanism explains it?
Level 3	Which constraint governs it?
Level 4	How do mechanisms interact?
Level 5	Which constraints compete?
Level 6	Can recovery occur?
Level 7	Which intervention should occur?
Level 8	Can the trajectory be controlled?
Level 9	Can adaptation be maintained?
Level 10	Can multiple systems be coordinated?

Each level expands the scope of structural reasoning while preserving the explanatory foundation established by mechanism attribution.

The maturity model therefore represents an evolution of benchmark targets rather than a hierarchy of benchmark difficulty.

F.13 Long-Term Research Questions

The roadmap naturally gives rise to several broader research questions.

Question 1

Can mechanism attribution be measured independently of outcome recognition?

Question 2

Can structural reasoning generalize across domains?

Question 3

Can intervention reasoning be benchmarked separately from prediction?

Question 4

Can closed-loop stabilization become an evaluable benchmark target?

Question 5

Can multi-agent structural reasoning be measured reliably?

Question 6

Can explanation support effective intervention?

Question 7

Can structural reasoning support closed-loop control?

Question 8

Can mechanism attribution scale to multi-agent environments?

These questions remain open.

The roadmap provides a structured path through which they may be investigated empirically.

F.14 Roadmap Summary

The Constraint Geometry framework introduced in this paper should be viewed as the first stage of a broader structural reasoning research program.

The benchmark family currently focuses on:

- Mechanism attribution
- Structural explanation
- Governing-constraint identification
- Failure reconstruction

Future benchmark layers may expand toward:

- Mechanism interaction
- Constraint competition
- Dominance switching
- Rescue geometry
- Intervention sequencing
- Closed-loop stabilization
- Adaptive compensation
- Multi-agent coordination

The long-term progression can be summarized as:

```
Outcome Recognition
         ↓
Mechanism Attribution
         ↓
Intervention Reasoning
         ↓
Trajectory Control
         ↓
Adaptive Coordination
```

Taken together, these extensions suggest a broader vision for benchmark evaluation.

Rather than evaluating only outcomes, future benchmark ecosystems may evaluate how models reason about:

- Structure
- Competition
- Intervention
- Control
- Adaptation
- Coordination

Whether these benchmark categories ultimately reveal distinct capabilities remains an empirical question.

The purpose of the roadmap is not to assume the answer.

It is to establish a path through which those questions can be investigated systematically.

The Constraint Geometry Benchmark Suite represents the first stage of that progression.