THE EFFECT PROPAGATION PROCESS (EPP): AN AXIOMATIC FOUNDATION FOR DYNAMIC CAUSALITY

Marvin Hansen
marvin.hansen@gmail.com
Date: September 24, 2025

ABSTRACT

Contemporary frameworks for computational causality provide a formal basis for inference within systems governed by static causal structures. These models, however, are predicated on assumptions of a fixed background spacetime and linear temporal progression, which become fundamental limitations when addressing complex dynamic systems where the causal laws themselves can evolve.

This monograph introduces the Effect Propagation Process (EPP), a formal meta-calculus for dynamic causality. It provides a single axiomatic foundation, a formal language, and a set of computable primitives for constructing, composing, and verifying domain-specific, causal theories. The EPP's single axiomatic foundation that causality is a spacetime-agnostic functional dependency necessitates a trifecta of core architectural components. The Context is an explicit and dynamic hypergraph that models the operational environment, supporting non-Euclidean and non-linear temporal structures. The Causaloid is a polymorphic, composable unit of causal mechanism that can encapsulate any specific object-level calculus. The Effect Ethos is a deontic guardrail that uses a defeasible calculus to ensure a system's actions remain verifiably aligned with its safety and mission objectives.

The EPP is fully implemented in the open-source DeepCausality peoject, demonstrating the work's practical application. Together, the EPP and its implementation in DeepCausality provide a foundation for the design, simulation, and implementation of a new generation of robust, context-aware, and verifiably aligned dynamic causal systems.

Contents

1 Introduction

The study of cause and effect has served humanity for millennia and provided a foundation for scientific inquiry and understanding. Contemporary frameworks for computational causality, from Pearl's Structural Causal Models to Granger's time-series analysis, provide a formal basis for causal inference within systems governed by fixed causal structures. These classical models, however, are predicated on a set of core assumptions, including a fixed background spacetime, linear temporal progression, and static causal structures. However, at the frontiers of science and engineering, a new category of challenges arises where the rules of causality themselves become dynamic. For these complex dynamic systems, the classical assumption of a static causal structure embedded in a fixed background spacetime is no longer applicable and imposes a fundamental limitation. From the dynamic regime shifts in financial markets with complex, multi-scale temporal feedback loops to context-dependent safety of autonomous vehicles, there is a need for a new foundation to model systems where causal structures themselves can evolve dynamically.

The presented monograph introduces the Effect Propagation Process (EPP), a single axiomatic foundation for dynamic causal models. Its purpose is to serve as a foundation upon which a new generation of domain-specific dynamic causal models can be built. Categorically, the EPP is closest to a meta-calculus because it defines the abstract mechanisms of dynamic causality while leaving the specifics to a derived dynamic causal model. However, the EPP also borrows from other categories:

- A Meta-Theory: The EPP provides the philosophical and logical foundation via its metaphysics and ontology upon which one can build a theory of dynamic causality.
- A Meta-Algebra: The EPP provides the formal, abstract language via its formalization of Causaloids, Contextoids, and their structural relationships.
- A Meta-Calculus: The EPP provides the formal, computational elements of dynamic causality via the Effect Propagation Process and the Deontic Inference Cycle.

The EPP adds dynamics as a first class principle to causality based on a single axiomatic foundation that generalizes causality as a spacetime-agnostic functional dependency. The operationalization of the EPP's single axiomatic foundation necessitates a trifecta of computable, first-class primitives: The Context, the Causaloid, and the Causal State Machine.

The detachment from a fixed background spacetime requires an explicit and dynamic Context. The EPP design of the context enables Euclidean and non-Euclidean representation and linear and non-linear temporal structures, thus supporting the modeling of rich, dynamic, and complex operational environments.

The functional nature of the axiom requires a polymorphic container for any specific causal calculus, the Causaloid. The causaloid, a concept borrowed from physicist Lucian Hardy, unifies cause and effect into a single abstract entity that solves a fundamental problem of structural composition by enabling isomorphic recursive causal structures. The EPP introduces three modalities of dynamic causality: dynamic, adaptive, and emergent. While dynamic and adaptive causality remain deterministic, the introduction of emergent causality, where causal structures co-emerge with their context, also introduces non-determinism with the implication that static verifiability is no longer possible. The Effect Ethos provides an operational guardrail to emergent causality based on a defeasible deontic calculus.

The Causal State Machine is a formal mechanism that translates causal reasoning into actions. A causal state machine separates its state from an action to allow an optional Effect Ethos to verify a proposed action against a set of norms to decide the permissibility of an action relative to the encoded ethos. The causal state machine combined with an Effect Ethos enable dynamic causal system in a dynamic environment while ensuring compliance as code to enforce alignment of derived actions.

1.1 Precedent

The presented Effect Propagation Process builds upon rich and diverse preceding scholarly work.

Whitehead's Process Philosophy

The EPP's primary departure point is a fundamental rejection of the classical Newtonian conception of a static, absolute background spacetime. This move is deeply rooted in the tradition of process philosophy, which argues that reality is not composed of enduring, static substances but is a dynamic flow of interconnected events. This idea finds its clearest expression in the work of Alfred North Whitehead, who posited a universe of "actual occasions"[1], and Henri Bergson, who described reality as a continuous "creative evolution"[2]. Their shared insight of reality as a process inspires the EPP's foundational redefinition of causality itself, shifting from a static, happen-before relation to a dynamic process of effect propagation.

Einstein's Theory of General Relativity

Einstein's theory of General Relativity[3], demonstrate that spacetime is a dynamic fabric, its geometry shaped through the gravity of the matter within it, which in turn influences the motion of matter. The EPP's concept of a Contextual Relativity that is both influenced by and influences the entities within is a direct analogue of this profound physical insight.

Pearl's SCM

Judea Pearl, with his SCM, established the foundation upon which the entire field of computational work was subsequently built. His foundational work in "Causality: Models, Reasoning, and Inference" [4] was as influential as his foundational critique in "Theoretical Impediments to Machine Learning with Seven Sparks from the Causal Revolution" [5]. In particular, his contribution to the algorithmization of counterfactuals has proven instrumental for the development of contextual counterfactuals.

Bareinboim's Transportability of Causal Effects

Bareinboim's calculus of transportability [6] and his subsequent work on data fusion formalize the very problem of contextual variance that the EPP's explicit context is designed to manage at a computational level. Where Bareinboim provides the definitive logical framework for reasoning about moving causality between discrete contexts, the EPP provides the computational primitive, the dynamic, queryable, and multi-modal Context, to operationalize this reasoning as part of the EPP.

Forbus's Defeasible Deontic Calculus

Kenneth Forbus's work on formalizing deontic calculus [7] has proven invaluable to solve the complex topic of conflicting norms in the effect ethos. In practice, it is rarely possible to write conflict-free norms, and during the development of the effect ethos, a recurring theme was the acceptance of this reality. The subsequent search for a solution led to the adoption of Forbus's Defeasible Deontic Calculus as the primary means to resolve normative conflicts.

Russel's Critique on Causality

Bertrand Russell critique on causality formulated in his 1912 essay "On the Notion of Cause"[8] directly lead to the realization that it's not necessarily causality that is central to Russels objection, but the underlying assumption of time asymmetry that is at odds with his observation that most successful theories in physics are based on time symmetry. Russel hinted at a profound truth because while physics routinely models dynamic change in complex systems, computational causality consistently struggles with capturing dynamic causality. While there is truth to causal invariance, there is also the reality that dynamic systems emit different causal structures depending on dynamic change and that is where computational causality is fundamentally at odds with physics and to an extent with reality. From there, it became clear that for causality to handle dynamics requires a new foundation of causality itself.

Hardy's Causaloid

Lucian Hardy introduced the "causaloid,"[9] a concept that encapsulates a spatial region and the causal connections within as a foundation for his work on finding a theory of Quantum Gravity. Critically, unlike all prior forms of causality, Hardy's causaloid is spacetime-agnostic because it folds cause and effect into one entity and thus removes the need for temporal order. His seminal work "Probability Theories with Dynamic Causal Structure" [10] had a threefold impact. First, during the foundational work of lifting causality into geometric structures, his causaloid formalism has proven instrumental in the formation of isomorphic recursive causal data structures. Second, his insight that the causaloid formalism puts deterministic and probabilistic structures on equal footing directly led to the multi-modal reasoning of the EPP. Third, his demonstration that fundamental differences of theoretical foundations are contained in a causaloid directly informed the representation of causal relations as a causal function, which then resulted in the single axiomatic formulation of the EPP.

Bornholdt's Uncertain Type

Bornholdt's et al. contribution in "Uncertain< T >: A First-Order Type for Uncertain Data" [11] directly informed the unification of deterministic and probabilistic reasoning in the EPP. Instead of representing a value with a single number, the Uncertain type in the EPP represents a probability with a full probability distribution (e.g., Normal, Bernoulli) or even a complex computation graph that produces a distribution. More importantly, the EPP reasoning logic can seamlessly "lift" simpler Deterministic and Probabilistic effects into the Uncertain distribution, aggregate all distributions, and then infer a logical combination of all the input distributions without loss of information. The final output is an Uncertain<bool> that collapses rich uncertainties into a single value at the very last moment while preserving second-order properties such as the standard deviation or confidence level. As a result, decisions, and crucially, deontic reasoning become more robust under uncertainty.

1.2 Contribution

The Effect Propagation Process contribution spans three distinctive areas. First and foremost a single axiomatic foundation formally derives the entire EPP from first principles. Second, the EPP establishes dynamic and emergent causal structures and an accompanying computable ethics framework for verifiable safety of dynamic causal systems. Third, a reference implementation in Rust demonstrates the feasibility of dynamic causality.

A Single Axiomatic Foundation of Dynamic Causality

The EPP's single axiom, causality as a spacetime-agnostic functional dependency generalizes causality so that the classical definition becomes a special case of it. Section 2 establishes the underlying philosophical foundation of the EPP. The details of the single axiomatic foundation are established in section 4.1, the implications of the generalization are discussed throughout section 5 and 7, and the formalization substantiates the EPP in section 9 further. The generalization is further substantiated in section 4.12 where five established methods of classical causality are expressed through the EPP and in Appendix A with a formal proof that the EPP subsumes the SCM. The SCM was chosen for the proof because of its historical and foundational significance to the field of computational causality. The reframing of causality as a single axiomatic functional dependency establishes explicit context and the causaloid as first-class structures. On important conjecture that follows from the axiom of functional dependency is that the field of function theory becomes applicable to causality as derived in section 4.1.3 and further discussed in section 11.1. The impact of the new foundation results in a single, coherent language to model advanced dynamic causality in physics, software, finance, and system biology with equal rigor.

A Formalism for Dynamic and Emergent Causal Structures

The EPP is designed form the ground up to model meta-causality, or the causality of causal change. The EPP defines an operational Generative Function is a function that can modify the causal model, its context, or both dynamically to describe causal emergence. Section 5 establishes the foundation of causal emergence, the higher-order implications are discussed in section 7, and section 9.7 establishes the formalization. The impact of causal emergence enables systems that can generate novel causal strategies in response to unforeseen circumstances. When safeguarded by the Effect Ethos, this provides a path to creating robust and resilient systems that can safely navigate a dynamically changing world.

A Computable Ethics Framework for Verifiable Safety

Traditional causal models are overwhelmingly focused on passive inference and prediction. They provide a powerful way to answer "what if" questions but lack a formal, verifiable, and safe bridge from that knowledge to taking action. Safety, ethics, and alignment are treated as post-hoc problems to be solved with testing and ad-hoc rules.

The EPP is the first computational causality framework to treat verifiable safety as first-class formal primitives directly integrated into the foundation itself. The motivation for the Effect Ethos is rooted in safeguarding causal emergence, and the EPP achieves this with two mechanisms:

- The Causal State Machine (CSM): Links a causal inference to a deterministic action.
- The Effect Ethos: The Effect Ethos allows a system to formally verify that a proposed action is compliant with its mission and safety objectives before it is executed.

The impact of the Effect Ethos results in a feasible roadmap for building trustworthy, high-stakes autonomous systems that adhere to contextual rules of engagement encoded in an immutable ethos. The effect ethos inverts the entire verification and validation model by shifting the safety objectives of a system into the pre-design stage to convert existing rules of engagement into a testable effect ethos, which enables design-time safety and provable alignment.

A Reference Implementation in Rust

The presented Effect Propagation Process has been fully implemented in the opensource DeepCausality[1] project hosted at the Linux Foundation for Data & AI. The reference implementation contributes three sub-projects. UltraGraph, a high performance, two-phase hyper-graph data structure used in the DeepCausality project. Uncertain, an implementation of Bornholdt's Uncertain Type in Rust. DeepCausality, the reference implementation of the EPP. The five established methods of classical causality expressed through the EPP in section 4.12 are demonstrated as code examples[2] in the DeepCausality project. At the time of writing, the DeepCausality mono-repository has reached a total of fifty thousand lines of Rust code with a sustained three months average test coverage[3] of 95%.

---

[1] www.deepcausality.com

[2] https://github.com/deepcausality-rs/deep_causality/tree/main/examples

[3] https://app.codecov.io/gh/deepcausality-rs/deep_causality

1.3   Structure

The presented monograph can be read from multiple angles. For the reader with a background in Philosophy, a natural path is to begin with the philosophy of causality (Chapter **??**) and the overview of the EPP (Chapter 4), and then delve into its philosophical foundation: the Metaphysics (Chapter 5), the Epistemology (Chapter 6), and the Teleology (Chapter 7).

For the reader with a background in Formal Methods, the EPP's core concepts are in Chapter 4), then the formal Ontology (Chapter 8), the Formalization of the meta-calculus (Chapter 9), and its soundness in Chapter **??**.

A reader with a background in Engineering may start with the Motivation (Chapter **??**), understand the high-level EPP framework (Chapter 4), and then see how these concepts are applied in the DeepCausality Implementation (Chapter 10). The Metaphysics in (Chapter 5) provides details about the orthogonal design used throughout the implementation.

The reader with a background in Computational Causality may start with the critique of classical models in the Motivation (Chapter **??**) and the Related Work (Chapter 3) is a good start. Following the EPP's core concepts (Chapter 4), the next step is the Formalization (Chapter 9) and validity (Chapter **??**). From there, the Future Work (Chapter 11) will be of particular interest.

2 Philosophy of Causality

2.1 The Classical Philosophy of Causality

2.1.1 Plato

Plato is believed to be the first to have explored the cause in a systematic way, most notably in his dialogue Timaeus[12] (c. 360 BC). In this work, Plato explains the creation of the cosmos through the actions of a divine craftsman, the Demiurge. The Demiurge looks to the eternal Forms as a model to bring order to the pre-existing, chaotic matter. Plato's concept of causality is thus tied to his broader metaphysical theory of Forms, where true causes are the eternal patterns or blueprints of which the physical world is merely a copy. He identifies several contributing factors necessary for the creation of the world, including the Demiurge (the efficient cause), the Forms (the formal cause), and the Receptacle (the material space)[12].

2.1.2 Aristotle

A student of Plato, Aristotle (c. 350 BC) formalized the notion of causality in his Metaphysics[13] with the "Four Causes"[14]:

1. The material cause or that which is given in reply to the question, "What is it made out of?"

2. The formal cause or that which is given in reply to the question, "What is it?". What is singled out in the answer is the essence of the what-it-is-to-be something.

3. The efficient cause or that which is given in reply to the question, "Where does change (or motion) come from?". What is singled out in the answer is the whence of change (or motion).

4. The final cause, the end purpose, is given in reply to the question, "What is its good?". What is singled out in the answer is that for the sake of which something is done or takes place.

Aristotle's framework provided a comprehensive vocabulary for analyzing causality that became foundational to Western scientific and philosophical thought for nearly two millennia.

2.1.3 Seneca

Seneca (c. 56 AD) argues in Letter 65[15] that cause and effect operate within a stage (space) and follow an order (time). Remove the stage or the order, and the conventional understanding of 'making something' or 'causing something' breaks down. His argument highlights time and space as indispensable prerequisites for classical causality. His focus on space and time as necessary conditions served as a precursor for physical concepts that treat spacetime as a background for causal processes.

2.2 The Realist View: Discovering Causal Laws

The realist tradition posits that causal relationships are objective, structural features of the world to be discovered.

2.2.1 Gottfried Wilhelm Leibniz

The idea of space and time as a background for causality, however, did not remain unchallenged. Gottfried Wilhelm Leibniz (1646–1716) rejected the concept of absolute space and absolute time as independent, fundamental constructs. Instead, Leibniz proposed[16] a relational view in which space is the simultaneous relation of coexisting things and time is the relational order of successive things. Through rigorous first principles analysis, Leibniz argued that the concept of absolute space and time was logically untenable. His relational perspective offered a significant alternative to the preeminent Newtonian worldview of his time.

2.2.2 John Stuart Mill

John Stuart Mill (1806-1873) shifted the focus of causality from metaphysical inquiry to a more practical, methodological problem. In his influential work, *A System of Logic, Ratiocinative and Inductive*[17], Mill developed a set of principles, now known as "Mill's Methods," to serve as a guide for discovering causal connections through empirical observation. These methods are designed to systematically eliminate non-causal factors and isolate true causes. Mill's work provided a formal basis for the inductive reasoning that underpins modern experimental science.

2.2.3 Albert Einstein

Albert Einstein (1879–1955) departed from Newtonian physics with his theory of general relativity[3] (GR), in which he established that space and time are one manifold, spacetime, that is bent by the gravitational influence of large masses. General relativity preserves the prerequisite of a spatiotemporal context for causality, echoing Seneca's key insight. However, the notion of a dynamic spacetime requires a dynamic view of causality to fit into the dynamic spacetime manifold.

2.3 The Empiricist View: Constructing Causal Models

The empiricist tradition, born from a deep skepticism, argues that causality is not a directly observable force. Instead, it is a mental model that we, as observers, construct to make sense of the statistical regularities we perceive in the world.

2.3.1 David Hume

The Scottish philosopher David Hume (1711–1776) launched a more fundamental attack on the notion of causality itself. In his A Treatise of Human Nature[18], Hume argued that we can never have direct empirical evidence of a necessary connection between a cause and an effect. All we can observe is their constant conjunction in space and time. The idea of "causality" as an unseen force is, for Hume, a product of the human mind develop after observing repeated patterns. Hume's skepticism decouples the idea of causality from any metaphysical necessity and re-grounds it in observational patterns[18].

2.3.2 Immanuel Kant

The German philosopher Immanuel Kant (1724 - 1804) agreed with Hume's premise that causality is not something we can empirically observe in the world. However, he disagreed with Hume's conclusion that it is merely a habit of the mind. In his Critique of Pure Reason [19], Kant argued that causality is an a priori category of understanding. It is a fundamental, built-in "rule" that the mind itself imposes on the raw data of sensory experience to make that experience coherent and understandable. We do not learn causality from the world; rather, we cannot experience the world in an intelligible way without already presupposing the principle of cause and effect. Kant reframes causality from an objective feature of the world (the Realist view) or a mere psychological habit (Hume's view) into a fundamental component of the cognitive architecture of any rational agent.

2.3.3 Hans Reichenbach

Building on Hume's empiricism, Hans Reichenbach (1891 - 19543) was among the first to propose a formal, probabilistic solution to the problem of induction. He argued that causal relationships are not deterministic laws but statistical structures. His 'Principle of the Common Cause' provided the foundational logic for inferring causal structures from statistical correlations[20] that later became known as the Reichenbach's Principle used by Pearl to define the concept of a confounder in modern graphical causal models[4].

2.3.4 Karl Popper

The challenge of induction, as framed by Hume, was famously addressed by the philosopher Karl Popper (1902 – 1994). Popper argued that causal laws can never be verified by observation, only falsified. From this perspective, a causal theory is not a description of a hidden force, but a bold conjecture—a falsifiable model—that is held to be provisionally true only as long as it survives rigorous empirical testing[21].

2.4 The Practical View: Causality in Physics

2.4.1 Bertrand Russell

Bertrand Russell (1872–1970) observed that successful physics has its roots in sophisticated, law-based descriptions of how a system evolves dynamically. In modern physics, the focus is on the state of a system (e.g., position, velocity, field strength) and how that entire state evolves continuously and dynamically. Therefore, for Russell, the idea of classical causality, a strict happen-before relation, no longer matches the reality of modern physics. Consequently, Russell wrote in his 1912 essay "On the Notion of Cause"[8]: *The law of causality, [...], is a relic of a bygone age."* Many modern physics laws are time-symmetric, which means that if state S1 at time t1 is related to state S2 at time t2 by a law, it is equally true that state S2 at time t2 is related to state S1 at time t1. This relationship is not a simple, linear, one-way street from a necessary "cause" to a dependent "effect." Knowing the state at any time allows

to calculate the state at any other time, past or future. Therefore, which state is the "cause" and which one is the "effect" becomes arbitrary. Russell was not opposed to causality itself; instead, his primary argument was that the traditional philosophical interpretation of causality as a fundamental, temporally asymmetric, and directed link is not what Russell observed in physics. Russell saw physics as a discipline to find, formulate, and test time-symmetric laws of dynamic change.

2.4.2 Causality in Quantum Physics

Russels view of physics describing time-symmetric laws of dynamic change was taken one step further in quantum physics because there the understanding of causality evolved from a structure that required the pre-existence of space toward a dynamic generative process from which causality emerges[22]. This emergent causality does not rely on a pre-existing spacetime but is grounded in a set of underlying rules (i.e., conceptualized as a 'generating function') that determines the fundamental materialization of spatiotemporal properties. The conceptualization of this fundamental level as a "generating function" captures the idea of a quantum process from which the necessary condition of classical causality's spatiotemporal structure arises. It is a shift from asking, "What causes X, given spacetime?" to "What process generates spacetime (and thus enables X to be caused)?".

2.5 The Functional View of Causality

The preceding philosophical traditions, from the Realists to the Empiricists, share a common language rooted in the concepts of objects, causes, and effects. Their view differ in whether these concepts correspond to objective reality or are mental constructs, but the concepts of causality themselves remain the same. The practical view from physics, as articulated by Bertrand Russell, suggests that there might be a third view of causality. Russel correctly observed that the lack of time symmetry disqualifies causality from its application to modern physics that models dynamic change. And indeed, causality does struggle with complex dynamic because of the inherent assumption of linear and asymmetric time. The author agrees with the premise and therefore generalizes causality as a spacetime agnostic functional dependency.

This functional view of causality is formalized in the EPP's single axiom from which the entire framework is derived:

$$E_2 = f(E_1)$$

The functional view redefines causality in the most general terms, the relationship between an input effect (E1) and an output effect (E2) via a transformation denoted as a function f(). The full reasoning behind the functional view and that, in fact, it subsumes the classical definition of causality is elaborates in section 4. From a philosophical perspective, the functional view of causality lead to a cascade of consequences:

1. Inherently Agnostic: The nature of the function f and the effects E are unconstrained and thus agnostic.
2. Inherently Computable: Functions compute and thus can be programmed.
3. Inherently Composable: Category theory is well established for functional caomposition.
4. Inherently Dynamic: Functions and higher order functions enable dynamics.
5. Inherently Formalized: Function theory has been studied for over a century and is well understood.

The realistic view of causality as an objective reality remains valid and yet compatible with the functional view because, from a realistic perspective, a function is nothing more than an objective, structural feature of the world. The empiricist view of causality as a model inferred from observed data also remains valid because, from the empirical perspective, the function is nothing more than the formal representation of the causal model derived from data. The practical view, from physics, however, stands to gain because here the function can represent an arbitrarily complex and dynamic physics process without any assumption of time. Within the EPP, one causal function might be deterministic, another one probabilistic, and yet another one describing a complex physical model using differential equations and yet all can be expressed using the functional foundation of the EPP.

The rest of the presented monograph explores the higher order effects that emerge from the functional view of causality. It begins with a review of the existing work in computational causality in section 3 and proceeds in section ?? with establishing a practical motivation for dynamic causality. The main contribution starts with rebuilding causality from first principles on a new functional foundation in section 4. Because of the introduced dynamics, a metaphysics discusses the necessary modalities to structure dynamic causality in section 5. Dynamic causality, however introduces a set of profound challenges that are discussed in the subsequent epistemology (section 6), Teleology (section 7) and Ontology (section 8). Section 9 then formalizes the effect propagation process that is derived from its single axiomatic foundation.

3   Related Work

The study of causality and causal inference aims to distinguish genuine cause-and-effect relationships from mere associations. Traditionally, establishing causality often relied on carefully controlled randomized controlled trials. However, significant theoretical advancements have shown that causal knowledge can be inferred from observational data by examining patterns of conditional independence among variables, given explicit assumptions [5].

3.1   Foundational Theories of Causal Inference

The endeavor to formalize and compute causal relationships draws upon several influential theoretical frameworks. Understanding these foundations is crucial for situating contemporary advancements and appreciating the nuances of different approaches to causal reasoning.

3.1.1   Directed Acyclic Graphs (DAGs)

A foundational framework for representing causal structures is based on graphical causal models, most notably Directed Acyclic Graphs (DAGs) [23, 24, 25, 26]. In these models, variables are typically represented by nodes, and directed edges indicate direct causal influences [24]. The impact of interventions, conceptualized by operators like the do-operator which sets a variable's value independently of its usual causes, can be analyzed within this framework to predict outcomes under hypothetical scenarios [24, 27]. The theoretical underpinnings of Structural Causal Models (SCMs), which are closely related to graphical models, have been extensively studied [4, 28, 29, 30, 31]. Methods exist for handling complex scenarios, including incorporating latent variables [32, 33] and understanding the relationship between different causal models [34, 5]. Policy interventions in specific graphical structures, such as Lauritzen-Wermuth-Freydenburg (LWF) latent-variable chain graphs, have also been investigated [35]. This includes work providing a novel identification result for effects of policy interventions in these graphs [35].

3.1.2   Structural Causal Model (SCM)

The dominant paradigm in modern computational causality is arguably the Structural Causal Model (SCM) framework, extensively developed by Judea Pearl and his colleagues [27]. An SCM consists of a set of variables, some of which are designated as exogenous (external, uncaused within the model) and others as endogenous (their values are determined by other variables within the model). The relationships between these variables are represented by a set of structural equations, typically of the form $X_i = f_i(\mathbf{PA}_i, U_i)$, where $\mathbf{PA}_i$ are the direct causal parents of X_i in the associated causal graph, and U_i are exogenous error terms representing unmodeled influences or inherent stochasticity. These structural equations are considered to represent autonomous, invariant causal mechanisms. Graphically, SCMs are typically depicted using Directed Acyclic Graphs (DAGs), where nodes represent variables and directed edges $X_j \rightarrow X_i$ indicate that X_j is a direct cause of X_i (i.e., $X_j \in \mathbf{PA}_i$). This graphical representation provides an intuitive way to encode causal assumptions and to determine statistical independencies via the criterion of *d-separation*.

A cornerstone of Pearl's framework is the *do-calculus*, a set of three axiomatic rules that allows for the inference of the effects of interventions from a combination of observational data and the causal graph structure, even when direct experimentation is not possible [27]. An intervention, denoted $do(X_j = x'_j)$, represents an external manipulation that sets the variable X_j to a specific value x'_j, thereby severing the links from its original parents $\mathbf{PA}_j$ and altering the system's natural dynamics. The ability to calculate post-intervention distributions, $P(Y|do(X = x))$, is central to predicting the consequences of actions and policies. Bayesian Networks, which are DAGs coupled with conditional probability distributions $P(X_i|\mathbf{PA}_i)$, are closely related to SCMs and are often used to represent the observational probability distribution $P(\mathbf{X})$ entailed by an SCM under specific assumptions about the error terms U_i. They provide a powerful tool for probabilistic inference under passive observation, but require the *do-calculus* or similar interventional logic to reason about causal effects.

15

3.1.3 Counterfactuals: What if?

While discovering causal structures and predicting the effects of interventions("What if we do $X = x$?") are fundamental tasks in causal inference, the ability to compute counterfactual queries represents a deeper and often more insightful level of causal reasoning [27]. Counterfactuals address questions about alternative realities or "what might have been"("What if X had been x', given that we observed $X = x$ and $Y = y$?"). This form of reasoning is crucial for tasks such as understanding individual responsibility, learning from past mistakes, diagnosing failures, and fine-tuning policies. It requires moving beyond population-level effects of interventions to consider specific individuals or units in specific factual circumstances [27, 36].

Within Pearl's Structural Causal Model (SCM) framework, computing a counterfactual, denoted as $Y_x(u)$ (the value Y would have taken in unit u had X been x), involves a three-step algorithmic process [27]:

1. **Abduction:** Use the available factual evidence (e.g., observed values of some variables) to update the probability distribution over the exogenous variables U. This step accounts for the specific unit or situation under consideration by inferring the background conditions consistent with the observed facts.

2. **Action:** Modify the original SCM by replacing the structural equation for the counterfactual antecedent X with $X = x'$(the hypothetical condition), effectively performing a "mini-surgery" on the model as in the *do*-calculus. The equations for other variables remain unchanged, reflecting the principle that interventions only alter the targeted mechanism directly.

3. **Prediction:** Compute the probability of the counterfactual consequent Y using the modified model and the updated distribution of U (from the abduction step). This yields the probability $P(Y_{x'} = y'|\text{evidence})$.

This process allows for a principled way to reason about hypothetical scenarios that differ from what was actually observed, effectively comparing parallel possible worlds [27, 37]. Foundational work also explored the bounding and identification of specific types of counterfactual queries related to probabilities of causation [38]. Formal systems like Pearl's *do*-calculus provide tools for determining if causal effects under intervention are identifiable from observational data [39], and algorithms exist to automate this process [40], which are often prerequisite steps before full counterfactual queries can be comprehensively addressed.

The extension of these concepts to practical applications and more complex settings remains an active area of research. For instance, model-agnostic approaches aim to enable counterfactual reasoning without full specification of the SCM, particularly in dynamic environments where systems evolve over time. Furthermore, the domain of causal bandits, which focuses on online decision-making and learning under uncertainty, increasingly incorporates causal background knowledge and aspects of counterfactual reasoning to optimize sequences of actions and learn policies more efficiently than purely correlational reinforcement learning approaches [41, 42, 43, 44]. The capacity for counterfactual reasoning thus forms a critical component of advanced intelligent systems that can not only predict and act, but also reflect, learn, and adapt based on a deep understanding of cause and effect in alternative scenarios.

More recent work explores model-agnostic approaches to counterfactual reasoning, particularly in dynamic environments [45], and investigates optimizing treatment effects in such settings [46]. Causal bandits also incorporate causal background knowledge into online decision-making problems [41, 42, 43, 44].

3.1.4 Potential Outcomes

Alongside SCMs, Potential Outcomes Framework, also known as the Rubin Causal Model (RCM) [47], offers another rigorous foundation for causal inference, with early conceptualizations by Neyman [48] and formally developed for observational studies by Rubin [49]. It has been particularly influential in statistics, econometrics, and the social sciences. This framework defines the causal effect of a treatment (or exposure) on an individual unit by considering the potential outcomes that unit would exhibit under different treatment assignments. For a binary treatment $T \in \{0, 1\}$, each unit i is conceptualized as having two potential outcomes: $Y_i(1)$, the outcome if unit i receives the treatment, and $Y_i(0)$, the outcome if unit i receives the control. The individual treatment effect (ITE) is then $Y_i(1) - Y_i(0)$. A core challenge, often termed the "fundamental problem of causal inference," is that only one of these potential outcomes can be observed for any given unit [47].

Inference in this framework hinges on crucial assumptions, such as the Stable Unit Treatment Value Assumption (SUTVA), which posits no interference between units and well-defined treatment versions. When all confounders are believed to be observed, the key assumption is **ignorability** (or unconfoundedness), which states that treatment assignment is independent of potential outcomes, conditional on the observed covariates [50]. Under this assumption, causal effects can be estimated using methods like matching, stratification, or inverse probability weighting based on propensity scores [50].

In many settings, the belief that all confounders have been measured is not plausible. To address this, applied researchers have developed a powerful toolkit of **quasi-experimental methods** that can enable causal identification even in the presence of unobserved confounding. These methods are cornerstones of modern computational causality and include:

- **Instrumental Variables (IV):** This technique is used to handle unobserved confounding by leveraging an "instrument": a variable that is correlated with the treatment but is not causally related to the outcome except through its effect on the treatment. The instrument provides a source of exogenous variation in the treatment, allowing for the estimation of a causal effect that is not biased by the unobserved confounders [51].

- **Difference-in-Differences (DiD):** Leveraging panel data (observations of the same units over time), DiD estimates the effect of a treatment by comparing the change in the outcome for a treated group before and after an intervention to the change in the outcome for an untreated group over the same time period. This method controls for unobserved confounders that are constant over time by differencing them out[52].

- **Regression Discontinuity Design (RDD):** RDD is applicable when the treatment is assigned based on a sharp cutoff in a continuous variable (the "running variable"). By comparing the outcomes of units just below the cutoff to those just above it, RDD can provide an unbiased estimate of the local causal effect at the threshold, mimicking a randomized experiment in a narrow window around the cutoff [53].

While SCMs provide an explicit language for encoding causal mechanisms, the Potential Outcomes framework, supported by these robust quasi-experimental methods, provides a powerful applied toolkit for estimating causal effects from complex observational data.

3.2 Invariant Prediction and Out-of-Distribution Generalization

A central challenge in modern machine learning remains robust out-of-distribution (OOD) generalization, as models trained under the standard I.I.D. assumption often fail when deployed in new or shifting environments. A powerful approach to this problem is rooted in the causal principle of invariance: the idea that while statistical correlations can be spurious and brittle, true causal mechanisms remain stable across different contexts [4].

This principle has been operationalized into a formal framework for both causal discovery and robust prediction. The seminal work in this area is Invariant Causal Prediction (ICP), developed by Peters, Bühlmann, and Meinshausen [54]. The ICP framework leverages data from multiple distinct "environments" or "settings." It posits that a set of variables constitutes the direct causes of a target if and only if the conditional distribution of the target given those variables remains invariant across all environments. By searching for a set of predictors that yields such a stable predictive model, ICP can identify causal relationships and produce a model that is robust to the types of distributional shifts observed during training.

This idea has been influential in the deep learning community, inspiring methods aimed at learning invariant representations. A prominent example is Invariant Risk Minimization (IRM), which seeks to learn a data representation such that the optimal classifier on top of that representation is the same for every training environment [55]. The goal is to isolate invariant causal features from spurious, environment-specific correlations. Other related approaches, such as Risk Extrapolation (REx) [56], also aim to improve OOD performance by enforcing penalties on models whose performance is unstable across environments. Collectively, this body of work formalizes the intuition that a model based on causal structure should generalize better than one that merely interpolates the training data, representing a major school of thought in building more reliable and robust machine learning systems.

3.3 Causal Discovery

A central task in the field is causal discovery, which focuses on learning the causal structure, represented by the graph, from observed data alone. A comprehensive survey categorizes existing methods for causal discovery on both independent and identically distributed (I.I.D.) data and time series data, including approaches for both types of data. According to this survey, categories include Constraint-based, Score-based, FCM-based, Hybrid-based, Continuous-Optimization-based, or Prior-Knowledge-based. Constraint-based methods infer relationships by testing for conditional independencies in the data [25, 57, 58]. Score-based methods search over potential graph structures and evaluate them based on how well they fit the data, often including a penalty for complexity [59]. The KGS method [60], for example, leverages prior causal information such as the presence or absence of a causal edge to guide a greedy score-based causal discovery process towards a more restricted and accurate search space. It demonstrates how incorporating different types of edge constraints can enhance both accuracy and runtime for graph discovery and candidate scoring, concluding that any type of edge information is useful. This method relates to the KCRL framework [60]. Continuous optimization techniques formulate causal discovery as an optimization problem, potentially involving differentiable approaches that can handle constraints like acyclicity. The NOTEARS framework is one such example [61], and studies have analyzed its performance and proposed post-processing algorithms to enhance its precision and efficiency. A study provides an in-depth analysis of the NOTEARS framework for causal structure learning, proposing a local search post-processing algorithm that significantly increased the precision of NOTEARS and other algorithms [62].

3.4 Causal Inference and Discovery for (Hyper)graphs

The representation of causal relationships via graphical models, predominantly Directed Acyclic Graphs (DAGs) as foundational to Structural Causal Models (SCMs) [27], is a cornerstone of computational causality. Much research has focused on discovering these graph structures from observational or interventional data (causal discovery) and subsequently estimating causal effects based on the identified graph (causal inference). While traditional methods often assume simpler pairwise relationships, the inherent complexity of many real-world systems necessitates considering more intricate relational structures.

Recent work has begun to explicitly tackle causal inference in settings involving multi-way interactions best represented by hypergraphs. Ma et al. [63] directly address the problem of estimating Individual Treatment Effects (ITE) on hypergraphs, specifically accounting for high-order interference where group interactions (modeled by hyperedges) influence individual outcomes. Their proposed HyperSCI framework leverages hypergraph neural networks to model these spillover effects and uses representation learning to control for confounders, demonstrating the utility of explicitly considering hypergraph topology for ITE estimation from observational data. This represents a significant step beyond assuming only pairwise interference, which is common in ordinary graph-based causal inference. While the work by Ma et al. focuses on statistical ITE estimation on a *given* hypergraph, it highlights the increasing recognition of hypergraph structures as vital for certain causal problems.

The broader field of graph mining and network science also provides a rich backdrop, with techniques for link prediction and understanding influence spread, though these often operate at a correlational level rather than a strictly causal one. The challenge remains to bridge network science concepts with formal causal reasoning in these complex relational systems.

3.5 Causal Inference for Time Series

Causal inference for time series data introduces a unique set of challenges and opportunities compared to static, cross-sectional settings. The inherent temporal ordering of observations provides strong, intuitive information about potential causal directionality—causes generally precede their effects—but also necessitates methods that can handle auto-correlation, non-stationarity, feedback loops, and varying time lags in causal influences.

A foundational concept in this domain is Granger Causality, originally developed by Clive Granger for economic time series [64, 65]. A time series X_t is said to Granger-cause another time series Y_t if past values of X_t contain information that helps predict future values of Y_t beyond the information already contained in past values of Y_t itself. This is typically tested using vector autoregression (VAR) models and statistical tests on the coefficients of lagged variables [66, 67]. While widely applied, standard Granger causality is primarily about predictive improvement and may not always align with true mechanistic causation, especially in the presence of unobserved confounders, instantaneous effects, or non-linear relationships. Extensions and refinements have been developed to address some of these limitations, including non-linear Granger causality tests and methods incorporating multivariate information criteria.

To explicitly model evolving causal relationships and dependencies over time, dynamic graphical models have been developed. The Dynamic Uncertain Causality Graph (DUCG) [68] is one such framework, specifically designed to represent and reason about causal relationships that themselves change as a system evolves. DUCGs find applications in complex dynamic systems, such as fault diagnosis in nuclear power plants where understanding the temporal progression of component failures is critical [69, 70]. These models often aim to unify diagnostic reasoning (what caused an observed state?) with treatment or control strategies (what intervention will lead to a desired future state?) [71].

More recently, deep learning techniques have been increasingly applied to causal discovery and inference in time series. For example, the Time-Series Causal Discovery Framework (TCDF) utilizes attention-based convolutional neural networks to learn causal relationships, explicitly trying to identify relevant time lags and dependencies [72]. Research in this direction often focuses on challenges such as optimizing hyperparameters for these complex models, ensuring robustness to varying noise levels and non-stationarities in the data, improving the interpretability of attention mechanisms to understand which past events are deemed causally salient, and developing robust causal validation methods beyond simple predictive accuracy. Other important research avenues in time series causality include:

- **Handling Unobserved Confounders:** Just as in static settings, unobserved common causes can induce spurious relationships between time series. Methods that attempt to detect or adjust for such confounding, perhaps using instrumental variable approaches adapted for time series or by searching for specific types of conditional independencies, are crucial.

- **State-Space Models and Causal Inference:** Integrating causal concepts with state-space models (e.g., Kalman filters and their non-linear extensions) allows for reasoning about causality between latent (unobserved) states as well as observed variables.

- **Interventional Time Series Analysis:** Developing methods to estimate the effect of specific interventions applied at certain points in time on the future trajectory of one or more time series. This is vital for policy evaluation and system control.

- **Causal Discovery from Irregularly Sampled or High-Dimensional Time Series:** Many real-world time series (e.g., medical patient data, sensor networks) are not regularly sampled or involve a very large number of variables, posing challenges for traditional methods.

- **Information-Theoretic Approaches:** Methods like Transfer Entropy [Schreiber, 2000, *Measuring information transfer*] provide a non-parametric way to quantify directed information flow between time series, offering an alternative perspective to Granger causality, especially for detecting non-linear interactions.

The temporal dimension thus adds significant complexity but also provides a powerful constraint (time ordering) that can be leveraged for causal reasoning, making this a vibrant and critical area of ongoing research.

3.6 The Role of Context in Causal Inference

While foundational causal frameworks like SCMs implicitly allow for conditioning variables, the explicit, structured, and dynamic modeling of *context* as a multi-faceted entity is a growing area of focus, crucial for applying causal inference to complex, real-world systems. The Jiao et al. survey [73] highlights numerous deep learning applications where contextual understanding is paramount, from visual commonsense reasoning to multimodal interactions, and notes the challenges posed by contextual shifts and confounders.

Berrevoets et al. [74, 75] propose a conceptual "map of causal deep learning" (CDL) that explicitly incorporates dimensions for structural knowledge, parametric assumptions, and significantly, a temporal dimension. They argue that time is not merely another variable but introduces unique considerations in causal settings, such as the fundamental principle that causes precede effects, and the potential for feedback loops or evolving relationships in dynamic systems. Their framework aims to help researchers and practitioners categorize CDL methods based on how they handle these dimensions, including whether they operate on static data or explicitly model temporal dynamics. For instance, they differentiate models based on whether they assume "no structure," "plausible causal structures" (often derived from statistical independencies), or a "full causal structure" as input, and similarly categorize parametric assumptions from non-parametric to fully known factors. The temporal axis distinguishes between static models and those designed for time-series data where variables are observed repeatedly. While this work by Berrevoets et al. primarily offers a *taxonomy and conceptual guide* for the emerging field of CDL rather than a specific implemented reasoning engine, it underscores the increasing recognition of structured context, and especially temporality, as a first-class concern in bridging deep learning with robust causal inference. This emphasis on temporal context aligns with established work in time series causality, such as Granger causality [64] and dynamic graphical models like DUCGs [68], which inherently focus on how relationships evolve over time. However, modern approaches, including those at the intersection of deep learning and causality, seek richer representations of temporal context beyond simple lagged variables. For example,

methods like TCDF [72] attempt to learn relevant temporal dependencies and attention patterns. The challenge remains to develop frameworks that can uniformly reason over diverse types of contextual information (static attributes, explicit temporal sequences, spatial relationships (both Euclidean and non-Euclidean), and even abstract conceptual states) and integrate this rich contextual understanding directly into the causal reasoning process. The ability to model multiple, potentially interacting contexts, and to allow these contexts to be dynamically updated, is key to building causal AI systems that can adapt to real-world complexities.

3.7 Hierarchical Causality

A significant frontier in causal inference is its extension to handle hierarchical or nested data structures, a common feature in fields from social science to biology. While hierarchical Bayesian modeling is a standard statistical tool for such data, causal modeling has traditionally forced a difficult choice: either aggregate the data to the unit level, losing valuable information, or ignore the group structure, risking incorrect inferences. Recent work by Weinstein and Blei[76] formalizes this problem by introducing Hierarchical Causal Models (HCMs). They extend Structural Causal Models (SCMs) by incorporating the concept of plates from graphical modeling to explicitly represent the nested structure. The central and powerful insight of their work is that disaggregating data and modeling the hierarchy can enable causal identification even in situations where it would be impossible with "flat" or aggregated data. For instance, with an unobserved unit-level confounder (e.g., a school's budget), the within-unit variation (e.g., student-level randomization) provides a "natural experiment" that can be leveraged to control for the confounder. To operationalize Hierarchical Causal Models, they develop a systematic, nonparametric identification procedure that extends the do-calculus. The HCM framework provides a formal causal justification for many existing methods, such as fixed-effects and difference-in-difference models, which can be seen as specific parametric instances of an HCM. Their work provides a broad and rigorous toolkit for analyzing cause and effect in multi-level systems, formally connecting the principles of hierarchical modeling with the inferential power of graphical causal models.

3.8 (Geometric) Deep Learning for Causal Inference and Representation

Deep learning has achieved remarkable success in various tasks, such as neural architecture search [77, 78] and techniques for handling complex relationships in data, such as those explored using hypergraphs [79, 80]. Hypergraphs, introduced by Berge in 1973 [80], can model multi-way relationships and have found applications in areas like visualization [81, 82], partitioning [83, 84, 85], and recommender systems [86, 87, 88, 89, 90, 91]. Link prediction, particularly in multiplex networks, is another active area where deep learning is applied [92, 93].

The intersection of deep learning with causal inference is a rapidly expanding research area, aiming to leverage the expressive power of neural networks to address challenges in causal representation learning, discovery, and effect estimation [94, 73]. Many approaches focus on adapting deep learning architectures to better estimate treatment effects from observational data, often by learning balanced representations of covariates to mitigate confounding bias or by modeling complex response surfaces. Ramachandra [95] proposes the use of deep autoencoders for generalized neighbor matching to estimate ITE, focusing on dimensionality reduction while preserving local neighborhood structure, and also suggests using Deep Neural Networks (DNNs) for improved propensity score estimation (PropensityNet). These methods exemplify the application of standard deep learning architectures to enhance specific statistical tasks within the potential outcomes framework, typically under assumptions such as the Stable Unit Treatment Value Assumption (SUTVA), which precludes interference.

A notable direction involves incorporating prior causal knowledge into deep generative models, enabling the generation of data that respects a given causal graph. While combining causal discovery with generative modeling is a goal, these methods are often constrained by the fundamental limitations of causal discovery [25]. Specific efforts include incorporating causal graphical prior knowledge into predictive modeling [96] and matching learned causal effects with domain priors in neural networks [97]. Applications in finance have also utilized informed machine learning frameworks based on a priori causal graphs for prediction tasks. The area of causal reinforcement learning and causal bandits also represents significant related work in combining causality with learning agents that interact with environments [41, 42, 43, 44, 98, 99].

More fundamentally, researchers are exploring how geometric deep learning principles can inform the design of causal models capable of handling complex data structures and respecting informational constraints. Acciaio et al. [100] introduce a "universal causal geometric DL framework," featuring the Geometric Hypertransformer (GHT). Their work is concerned with the universal approximation of causal maps between discrete-time path spaces, which may be non-Euclidean metric spaces such as Wasserstein spaces, while strictly respecting the forward flow of information inherent in causal processes. The GHT employs hypernetworks to adapt its parameters over time and aims to provide theoretical guarantees for approximating Hölder continuous functions between these complex spaces. Although highly theoretical and with a focus on applications in stochastic analysis and mathematical finance, this line of work signi-

fies a deep engagement with geometric structures, non-Euclidean spaces, and transformer-like attention mechanisms within a causal learning context. Their "geometric attention mechanism" operating on Quantizable and Approximately Simplicial (QAS) spaces represents a sophisticated approach to handling non-Euclidean output geometries. A survey by Jiao et al. [73] details various methods where deep learning is applied to causal discovery (e.g., leveraging neural networks for GraN-DAG or extensions of NOTEARS) or to augment specific causal inference tasks within existing deep learning modalities (e.g., developing causal attention mechanisms in computer vision, or applying causal methods to Graph Neural Networks). This body of work collectively seeks to imbue deep learning models with a degree of causal awareness or to use their representational power to overcome limitations in traditional causal inference techniques. The ongoing challenge is to move beyond enhancing specific sub-tasks towards building more integrated and principled causal deep learning architectures.

3.9   Causal Representation Learning

Causal Representation Learning (CRL) has emerged as a field dedicated to this problem, aiming to learn latent representations that are not just statistically useful but are also causally meaningful [101]. The central goal is to learn a mapping from high-dimensional observations to a latent space where the dimensions correspond to independent, underlying causal factors. Early efforts in deep learning focused on learning "disentangled" representations, with the hope that unsupervised methods could automatically separate data into its factors of variation. However, foundational work by Locatello et al. demonstrated that learning disentangled representations without inductive biases is theoretically impossible [102]. This finding has spurred the CRL community to propose that causal structure is the necessary inductive bias to achieve meaningful and identifiable disentanglement. The core assumption is that real-world changes often arise from interventions on a sparse set of high-level causal mechanisms. A model that captures these mechanisms in its latent space should therefore be more robust and generalize better out-of-distribution. A key technical challenge in this area is the identifiability of the latent causal variables—ensuring that the learned representation is unique and corresponds to the true underlying causal structure.

3.10   The Causaloid Framework

A significant effort towards establishing a framework for probabilistic theories with dynamic causal structure was presented by Hardy [10]. Hardy proposes a framework aimed at unifying quantum theory (QT) and general relativity (GR) as a step towards quantum gravity (QG). The core of this unification lies in a generalized theory of causality, capable of describing both the probabilistic nature and fixed causal structure of QT, as well as the deterministic nature and dynamic causal structure of GR, within a single formalism. The core of this new framework of unified causality is the "causaloid," a mathematical object designed to encapsulate all information about the causal relationships within a physical system. The framework begins from an operational standpoint, focusing on "recorded data" which consists of "actions" and "observations" associated with "elementary regions" of spacetime.

The causaloid itself is a theory-specific mathematical entity, primarily represented by a collection of "lambda matrices" (Λ). These matrices quantify how the complexity of describing a composite region (specifically, the number of fiducial measurements needed to determine its state) is reduced due to causal connections between its component elementary regions. Associated with any region R and an experimental procedure F_R resulting in outcome X_R are "r-vectors," denoted $r(X_R, F_R)(R)$, which are analogous to operators in QT [10]. A key innovation is the "causaloid product," which is governed by the causaloid (via the lambda matrices). This product combines r-vectors of sub-regions to form the r-vector for a composite region, e.g., $r(R_1 \cup R_2) = r(R_1)\hat{} r(R_2)$. This product aims to unify the different ways systems are composed in QT, such as tensor products for space-like separated systems and sequential (matrix) products for timelike evolutions. Probabilities for joint outcomes, conditioned on experimental settings, are then derived from these r-vectors. A crucial feature of the causaloid formalism is that it does not impose a fixed causal structure or a background time a priori. Instead, the causal relations are implicitly defined by the causaloid itself.

Hardy demonstrated how both classical probability theory and quantum theory can be cast within this framework, with the differences between theories being encoded entirely in the specification of their respective causaloids. The ultimate aim is to provide a structure wherein the dynamic causal aspects of GR can be consistently combined with the probabilistic nature of QT. Hardy also introduces "causaloid diagrams" as a visual tool to represent and compute the causaloid based on local lambda matrices for nodes (elementary regions) and links (pairwise connections), particularly under simplifying assumptions met by QT and classical probability [10].

3.11 Computational Causality Libraries

A vibrant ecosystem of Python libraries has emerged over time, providing tools for various aspects of causal inference, discovery, and analysis. These libraries typically build upon foundational causal theories and aim to make causal methods accessible to data scientists, researchers, and engineers. This report summarizes several key libraries shaping the Python landscape for computational causality.

3.11.1 DoWhy (Microsoft)

Developed by Microsoft Research, DoWhy[4] [103] is perhaps one of the most well-known libraries aiming to provide an end-to-end workflow for causal inference. Its philosophy centers on explicitly separating the causal modeling assumptions from the statistical estimation steps, adhering to a four-stage process: 1) Modeling the causal assumptions (often using graphical models), 2) Identifying the target causal estimand based on the model, 3) Estimating the causal effect using appropriate statistical methods (like propensity scores, regression, instrumental variables), and 4) Refuting the obtained estimate through robustness checks. DoWhy aims to unify concepts from both Pearl's Structural Causal Models (SCMs) and the Potential Outcomes framework. It integrates with other libraries like EconML and CausalML for specific estimation tasks and is designed to be a general-purpose tool for applied causal analysis.

3.11.2 EconML (Microsoft)

EconML[5] [104] focuses specifically on estimating heterogeneous treatment effects (HTE) – understanding how the effect of an intervention or treatment varies across different individuals or subgroups. It heavily leverages machine learning techniques to model complex conditional outcome expectations and propensity scores while incorporating causal identification strategies to ensure the validity of the effect estimates. Key methodologies implemented include Double Machine Learning (DML), Orthogonal Random Forests, Deep Instrumental Variables (DeepIV), and various "meta-learners" (S-learner, T-learner, X-learner) that adapt standard ML models for causal effect estimation. Meanwhile, DoWhy and EconML have been brought together in the PyWhy[6] project to foster industry wide collaboration.

3.11.3 CausalML (Uber)

Developed initially at Uber, CausalML[7] [105] is another library primarily focused on treatment effect estimation and, notably, uplift modeling. Uplift modeling specifically aims to estimate the incremental impact of an intervention on an individual's behavior – identifying who would be positively influenced by an action (e.g., receiving a promotion) compared to doing nothing. CausalML provides implementations of various uplift algorithms, including tree-based methods (causal trees/forests) and meta-learners similar to those in EconML. It's geared towards practical industry applications, especially in customer relationship management (CRM) and marketing, where optimizing interventions based on predicted individual uplift is a key objective.

3.11.4 CausalNex (McKinsey)

CausalNex[8] takes a different approach, focusing more strongly on causal discovery and the use of Bayesian Networks for causal reasoning. It provides tools to learn causal graph structures from data, potentially incorporating domain knowledge to constrain the search space. It implements structure learning algorithms (like NOTEARS) and allows users to fit Bayesian Networks to the data based on the learned (or provided) graph structure. Once the network is built, users can perform queries (e.g., conditional probability queries, interventions via the do-calculus if the graph assumptions hold) to understand relationships and simulate scenarios within the modeled system. CausalNex is particularly useful for exploring and visualizing complex systems where understanding the network of causal influences is a primary goal.

---

[4]https://github.com/py-why/dowhy
[5]https://github.com/py-why/EconML
[6]https://www.pywhy.org
[7]https://github.com/uber/causalml
[8]https://github.com/mckinsey/causalnex

3.11.5 Meridian (Google)

Meridian[9] is a marketing mix model (MMM) that uses Bayesian causal inference methods to offer better insights into online and offline marketing channels. Meridian provides methodologies to support calibration of MMM with experiments and other prior information, and to optimize target ad frequency by utilizing reach and frequency data.

3.12 Causal Inference with Large Language Models (LLMs)

The intersection of causal inference and Large Language Models (LLMs) has emerged as a vibrant and rapidly developing research frontier. Foundational work has mapped out the potential for using LLMs as interactive causal knowledge engines, capable of answering queries about causal relationships, interventions, and counterfactuals by drawing on the vast information embedded in their training data [106]. This opens up the possibility of automating parts of the causal modeling process that have traditionally been highly manual. However, a fundamental question shadows this potential: whether LLMs, trained on vast quantities of observational and correlational text, can distinguish true causation from spurious association. Research specifically investigating this issue has shown that LLMs often struggle to infer causation correctly when faced with scenarios where correlation and causation are deliberately misaligned, highlighting a significant risk of the models simply repeating the statistical patterns in their training data [107]. To address this challenge, the research community has focused on creating structured and comprehensive benchmarks to systematically assess the causal reasoning capabilities of LLMs. For instance, the CLADDER framework was developed to generate complex, language-based causal problems from underlying structural causal models, allowing for a controlled and fine-grained analysis of model performance and failure modes [108]. In a similar vein, CausalBench provides another comprehensive benchmark suite designed to evaluate a wide array of causal reasoning skills, from basic causal discovery to complex counterfactual inference [109]. Collectively, this body of work indicates that while LLMs show promise, they are not yet reliable causal reasoners.

3.13 In-context Causal Reasoning with Large Language Models

Recent advancements have seen the application of large-scale language models (LLMs) to tasks in causal inference, leveraging their ability for in-context learning. This emergent capability allows models to perform causal reasoning on the fly, based on the provided context, without requiring parameter updates. A notable direction of this research is in-context counterfactual reasoning. Miller, Schölkopf, and Guo[110] demonstrate that language models can perform counterfactual reasoning in a controlled synthetic environment. Their work suggests that for a wide variety of functions, counterfactual reasoning can be reduced to a transformation of in-context observations. The authors find that self-attention, model depth, and data diversity are key drivers of this capability in transformer architectures. This line of inquiry extends to sequential data, providing preliminary evidence for the potential of counterfactual story generation. Building on the concept of in-context learning, Schölkopf et al.[111] have proposed Do-PFN, a pre-trained foundation model designed to predict interventional outcomes from purely observational data. Their model is pre-trained on a wide variety of synthetic causal structures, which enables it to meta-learn how to perform causal inference. Do-PFN has shown strong performance in estimating causal effects, even without knowledge of the underlying causal graph, a significant departure from traditional methods that require such information.

3.14 Causal Inference at Industry Scale

An open challenge for the practical application of causal inference remains scalability. In enterprise environments such as Netflix, Google, and Meta, causal questions must be answered using datasets with millions or billions of observations. Causal inference at scale has exposed a significant gap between theoretical algorithms and practical feasibility due to two known limitations:

The first is causal discovery, where the goal is to learn the graph structure from data. Score-based search methods are generally NP-hard in the worst case [59], and even faster constraint-based methods like the PC algorithm face a combinatorial explosion of required conditional independence tests in dense or high-dimensional graphs [112].

The second is in causal effect estimation, especially in the presence of high-dimensional confounding. To address this, a significant body of work has emerged around Double/Debiased Machine Learning (DML) [113]. The DML framework provides a recipe for using powerful, arbitrary machine learning models to flexibly control for a large number of confounders without introducing bias into the final treatment effect estimate. This line of research is explicitly motivated by the need to apply causal inference in settings where the number of variables makes traditional statistical methods intractable. In response to these challenges, a major engineering focus has been the development of causal inference platforms. These internal systems are designed to automate and scale causal analyses, enabling data scientists to run thousands of experiments and quasi-experiments reliably and efficiently.

---

[9]https://github.com/google/meridian

3.15 Boundaries of Classical Causal Methods

The established methods of computational causality, particularly the frameworks developed by Pearl, Granger, and Rubin, represent monumental intellectual achievements that form the foundation of the field. The following assumptions are commonly shared by classical methods:

- Acyclicity (DAG): Causal relationships are acyclical in a directed acyclic graph.
- Static Causal Structure: The causal graph, once defined, is assumed to be static.
- Fixed background spacetime: Variables in the DAG are assumed to be within a fixed background spacetime.

Implications:

- **Non-Linear Time:** The acyclical assumption explicitly prohibits feedback loops, and the fixed background spacetime assumption prevents any form of non-linear time.
- **Non-Euclidean Representation:** The assumption of a fixed background spacetime assumes Euclidean structures and thus does not allow for non-Euclidean representation.
- **Non-Dynamic structures:** The assumption of static causal structure prevents dynamic causality.

While these assumptions enable powerful reasoning within their defined boundaries, they become untenable in complex dynamic systems, a view shared by DARPA:

> "In the real world, observations are often correlated and a product of an underlying causal mechanism, which can be modeled and understood."[114]

The problem is not a simple choice between correlation and causation. The deeper issue, as contemporary philosophers such as Luciano Floridi have argued, is that the predominant mental model underlying complex system design remains rooted in an outdated Aristotelian and Newtonian "Ur-philosophy" of fixed space and time [115].

3.16 Beyond Classical Causal Models

The need for the Effect Propagation Process arises directly from modeling complex dynamic systems, which are characterized by three deeply interrelated challenges: non-Euclidean representation, non-linear temporality, and dynamic causal emergence. The most critical challenge is dynamic causal emergence, where the causal laws of a system are themselves subject to change. This is a practical reality in numerous domains. In financial markets, for instance, the causal relationships between assets are relatively stable during normal conditions but undergo a fundamental "regime shift" following a black swan event or a major policy change.

In contemporary plasma fusion, it is well understood that the magnetic confinement field and the turbulent plasma it contains exist in a tightlycoupled feedback loop, where each dynamically influences the other in real-time. In such systems, the very nature of the causal relationships can change depending on the energy scale, a phenomenon known as "dynamic phase alignment", rendering classical acyclic or linear-temporal causal models insufficient[116].

Similarly, in climate science, ecological systems can cross "tipping points" where gradual changes trigger new, powerful feedback loops, fundamentally altering the causal structure of the climate model. This dynamic nature exacerbates the problem of non-linear time. In a system with emergent feedback loops, such as in climate modeling or high-frequency trading, the notion of a single, uniform temporal progression becomes untenable. Events on nanosecond, daily, and quarterly scales can all interact, forming a complex temporal hypergraph where causal influence is no longer follow a simple unidirectional time arrow.

Complex causal relations are often expressed through non-Euclidean representations. For example, a medical record documenting a patient's health cannot be reduced to a simple vector of measurements. It is a complex graph of relationships between genomic data, electronic health records, comorbidities, and environmental factors. Likewise, a global supply chain is not a rigid grid but a dynamic network of dependencies. Reducing these relational structures to a Euclidean vector space to fit classical models often destroys the very information needed for accurate causal inference.

These challenges, emergent causality, complex temporality, and non-Euclidean data, are not independent issues to be solved in isolation. They are deeply interrelated and jointly point to a single, underlying problem: modeling reality in complex, adaptive, and dynamic systems. This requires a fundamental rethinking of causality itself, moving away from a static, linear interpretation and towards a dynamic, context-aware, and dynamic process. More profoundly, a new foundation of causality becomes necessary from which contextual dynamic causal systems can be build upon.

4   Causality as Effect Propagation Process

The foundational premise of the EPP is the detachment of causality from a presupposed spacetime. This premise necessitates a re-evaluation of the causal relation itself, shifting its conceptualization to a more general process of effect propagation.

4.1   Definition

The notion of a generative process that underlies the fabric of spacetime leads to the implication that causality has to evolve beyond the strict "before-after" relation towards a spacetime-agnostic view. The critical question then becomes:

> "What is the minimum necessary and sufficient condition for a relationship to be considered causal,
> if one cannot use temporal precedence ('happen-before') in the definition?"

The answer to this question entails the very foundation upon which computational causality is built. The classical answer to this question for centuries has been: $Temporal\,Precedence + Counterfactual\,Dependence$. That means, the cause must come before the effect, AND the effect would not have happened without the cause.

The EPP, by its foundational act of detaching causality from spacetime, makes the "Temporal Precedence" condition untenable. It requires an answer that entails the essence of causality independent of spacetime. Finding that answer, however, starts with the classical definition of causality.

4.1.1   Classical Definition of Causality

The classical definition of causality, taken from Judea Pearl's foundational work[4]:

> IF (cause) A then (effect) B.
> AND
> IF NOT (cause) A, then NOT (effect) B.

When removing temporal order from causality, it is indeed no longer possible to discern cause from effect because, in the absence of time, there is no "happen-before" relation any longer, and therefore, the designation of cause or effect indeed becomes infeasible, just as Russell hinted at earlier on. When removing space from causality, the location of a cause or effect in space is not possible anymore because space itself is no longer available.

4.1.2   Deconstructing Classical Causality

The absence of spacetime raises the question: *What is the essence of causality?*

Logically, the answer comes in three parts:

1. Causality is a process.
2. Causality determines effects.
3. Causality describes how effects propagate.

The first one is self-explanatory because causality occurs in dynamic systems that change and therefore, causality must be a process. The idea goes back to Whitehead, has been further developed by Bergson, and finds precedent in Mill's method of concomitant variation[17].

The second one is less obvious, because one might think that causality is all about the "cause" that brings the effects into existence. However, let's think the other way around: We know that X is the cause of effect E, because E happens when X happens and because E does not happen when X does not happen either. Therefore, we can determine a cause in terms of its effects. An effect, in essence, is an observable change of state. Therefore, it is true that causality determines effects.

The third one, effect propagation, needs elaboration because it is commonly assumed that the cause is the dominant factor in causality. When we rewrite the previous definition of classical causality in terms of effect propagation, however, we see that there is no loss of information:

> If X happens, then its effect propagates to Effect E.
> AND
> If X does not happen, then its effect does not propagate to Effect E.

In this definition, X does not have a designated label and instead is described in terms of its emitting effect. Therefore, X can be seen as a preceding effect, which then propagates its effect further.

Thus, we can write without loss of information:

> If Effect E1 happens, then its effect propagates to Effect E2.
> AND
> If Effect E1 does not happen, then its effect does not propagate to Effect E2.

Therefore, causality becomes an effect propagation process. The effect propagation process definition is more general and treats the classical happen-before definition of causality as a specialized derived form. When the preceding effect is designated as a "cause", then you can rewrite the general definition back into the classical definition of causality, and therefore the generalized and the specialized definition of causality remain congruent. The EPP formalizes the 'effect' itself as a dedicated 'PropagatingEffect' together with supporting formalism to operationalize the effect propagation process.

4.1.3 Generalized Definition of Causality

The generalized definition of causality used by the EPP is:

> If Effect E1 happens, then its effect propagates to Effect E2.
> AND
> If Effect E1 does not happen, then its effect does not propagate to Effect E2.

The re-definition of causality as "effect propagation" is a more general, process-oriented term for functional dependency. To illustrate why that is the case, let's deconstruct the new definition:

1. "If Effect E1 happens, then its effect propagates to Effect E2." Here, E2 functionally depends on E1.
2. "If Effect E1 does not happen, then its effect does not propagate to Effect E2." The counter-factual case for E2 is equally functionally dependent on E1. That also means, a different input (NOT E1) necessarily leads to a different output (NOT E2), assuming the function is not a constant.

For the initial question, "What is the minimum necessary and sufficient condition for a relationship to be considered causal, if one cannot use temporal precedence ('happen-before') in the definition?", the answer is that the relationship is causal because one effect state is a function of another's. As a direct consequence of the generalized definition of causality, the single, minimum necessary and sufficient condition for a relationship to be causal is Functional Dependency.

4.1.4 Axiom of Causality

For the generalized definition of causality, the only applicable axiom is a spacetime-independent functional dependency between Effect E1 and Effect E2. Therefore, we can write:

E2 = f(E1)

It is important to recognize that the function f() and both effects are deliberately chosen to be unconstrained. That means, effects E1 and E2 might be deterministic, probabilistic, symbolic, or of arbitrarily complex structure. Furthermore, f() does not assume any particular modality either and thus could be a simple logical expression, a SCM, a DBN, a differential equation, or even a neural net. The only practical constraint is that the function must compute eventually. Application specific constraints can be imposed on the function and the effects, for example to support methods of formal verification.

4.1.5 Functional Conjecture

Based on the single axiom **E2 = f(E1)**, an important conjecture follows:

> let x = E1
> let y = E2
>
> **E2 = f(E1) <==> y = f(x)**

The axiom of causality as functional dependency directly leads to the application of the field of mathematical function theory and category theory. The application of category theory, for example, enables the formulation of a causal calculus, an idea further explored as Functional Causality in section **??**.

4.1.6 Higher-Order Implications

The higher-order implications of a single axiomatic foundation of causality entail, among others:

- If causality is functional dependency, then the fundamental unit of causality must be a container for a function. More precisely, the resulting unit will be equivalent to a higher-order function of $apply(f(E1))$.
- If the function needs inputs to operate on, then the system must provide those inputs.
- If the output of one function is to be the input of another, then it follows that input and output are isomorphic. It also follows that effects propagating from one causaloid to another, hence this is the materialization of the effect propagation process.
- If the function interacts with the outside world, then the system must provide the context.
- If the outside world changes, then the system must provide a mechanism to reflect those changes in the context.
- If these function-containers are to relate to one another, there must be a structure to hold them and define their pathways. This logically necessitates a causal graph structure.
- If the causal graph model complex, real-world systems, it must be scalable and composable.

The currently identified higher-order effects most likely will remain incomplete for the foreseeable future despite the EPP's attempt to describe and formalize an initial set. It is understood that the full exploration of the new possibilities that follow from the new axiomatic will take some meaningful time and effort.

The EPP accepts the premise of detaching causality from spacetime, it embraces causality as functional dependency as its core tenets and, as a result, many of the properties of the EPP are necessary consequences that arise from its single axiomatic foundation. This monograph explores the higher order implications listed in 4.1.6 and its application to computational causality.

4.2 Overview of the EPP

The starting point of the EPP is a generalized definition of causality as effect propagation, which detaches the concept from a presupposed spacetime. From the generalized definition of causality, a cascade of novelties followed to operationalize the EPP:

1. Explicit Assumptions
2. Externalized, Computable Context
3. Unified Causal Unit
4. Fractal, Self-Referential Causal Structure
5. Multimodal Propagating Effect
6. Causal State Machine
7. Teloid and Effect Ethos

Explicit Assumptions:

The EPP defines a "Model" as a set that comprises the core logic encapsulated in one or more causaloids, one or more contexts used by the causaloids, and a set of explicit assumptions that must hold true for the model to work. Conventionally, in classical causality, assumptions about the data are implicit and the decision whether a causal model can be transferred to a different dataset requires additional methods such as Invariant Causal Prediction (ICP) to determine if a causal model applies to a dataset with a different distribution. In contrast, the EPP elevates assumptions as a first-class entity that serves the purpose to decide upon model transferability by encoding assumptions into explicitly testable and computable units.

An Externalized, Computable Context:

The detachment from a fixed spacetime necessitates the externalization of the environment into a first-class Context. This Context is a rich and explicit representation of the world that the Causaloids read from and reason about. A

context may be a static representation of facts, a complex multi-scale temporal graph, a complex coordinate system, or any combination thereof. A static context emits an invariant structure after it is defined whereas in a dynamic context, the context structure itself evolves e.g. new elements are added or removed. In both cases, elements of a context can be adjusted and updated to reflect either new values or correction of existing values. The adjustment mechanism allows for error correction or the adjustment for relativistic effects.

Unified Causal Unit:

The EPP resolves the untenability of the classical 'cause-effect' separation by introducing the Causaloid. The Causaloid is a single, computable entity that unifies the mechanism of a cause with its effect. The Causaloid comprises a causal function and therefore establishes function theory and the related lambda calculus as the foundation for causality in the EPP.

Fractal, Self-referential Causal Structure:

The EPP inverts the classical relationship between a causal unit and its environment. In classical models, the causal structure is a pre-defined framework (such as a DAG) in which entities are placed. The EPP establishes a fundamentally entity-first, fractal, and self-referential definition of causality by defining the structure as relationships between causaloids and causaloids as the elements of the relationships. This mechanism is made operational through isomorphic recursive composition. The EPP establishes three distinct forms of causaloids that are all isomorphic, yet distinct: Singleton, Collection, Graph. The isomorphism allows a single Causaloid node to represent either a single causaloid, a collection of causaloids, or to encapsulate an entire, arbitrarily complex Causaloid Graph, enabling the concise expression of deeply layered systems. The collection exists as a specialized form of the graph to simplify common use cases without the complexity of a graph structure.

Multimodal Propagating Effect:

To facilitate reasoning within the causal structure, the EPP unifies causal input (Evidence) and output (Effect) into a single, isomorphic type: the PropagatingEffect. The Causaloid unifies cause and effect into one logical unit and the EPP is achieving this by formulating the Causaloid as a higher-order function, one that encapsulates a causal function and applies to data for reasoning. As a direct consequence, the distinction between causal input and output becomes quite arbitrary considering that the output of one causaloid is expected to be applied to another causaloid. Therefore, the deliberate decision was made to follow the same logic of the causaloid and fold causal input (Evidence) and output (Effect) into a single, isomorphic type that uniformly represent both and thus directly enable the application of causal output from one causaloid as causal input to another causaloid. This provides a single type that enables the principled unification of deterministic and probabilistic reasoning across arbitrarily complex causal structures.

Causal State Machine:

The EPP bridges the gap between causal reasoning and intervention with the causal state machine (CSM). The CSM uses a causal state that connects to the reasoning outcome via the PropagatingEffect and ultimately converts all complex reasoning into a binary outcome to indicate whether to act according to the defined causal action. When the causal state of a CSM evaluates to true, it then fires its causal action. The action is a regular function that may interact with the context, the causal model, or any external system.

Teloid and Effect Ethos

The clear separation between context, causal logic, and intervention has been designed to enable the interception of proposes actions for checks against operational rules. In regulated industries, an auditable trail for each action is legally required and by making the causal reasoning fully explainable before deciding upon an intervention directly supports internal auditing. Therefore, the EPP adds the Teloid as a single unit of intent that encodes a specific rule and the Effect Ethos that aggregates a set of teloids into a larger rule set. A causal state machine then uses the effect ethos to check if the action derived from the causal reasoning is compliant with the rules encoded in applicable teloids.

4.3 Explicit Assumptions

In the EPP, model assumptions are explicit and connected to the model. In practice, this allows to define a Model with a set of preconditions (the assumptions). Before using the model for reasoning or prediction, this allows to efficiently check if all its underlying assumptions hold true for the given data, ensuring the model is operating within its valid parameters. Furthermore, transferring a model to a new environment can be tested by sampling representative data and testing the samples against the model assumptions to gauge if a model transfer is feasible. In practice, even if an assumption test is positive, it is advised to test the model with an alternate context that closely resembles the new environment to gain more confidence in the degree of transferability.

4.4 Context

A key contribution of the EPP is the externalization of context as a first-class entity. The context of a causal model is a hypergraph that encapsulates supporting data. Each node in this hypergraph is a Contextoid, a unit of information that can represent:

- Data
- Time, Space, and Spacetime
- Symbol

The causal logic is kept distinct from the contextual data it operates on. It also directly enables the agnosticism to the structure of space and time, accommodating Euclidean, non-Euclidean, and symbolic representations within the same architecture. Furthermore, causal logic may operate on one or more contexts and, equally important, a particular context might be shared between different causal logic, thus enabling efficient and scalable context representation in complex dynamic systems.

4.5 Causaloid

In the Effect Propagation Process, due to the detachment from a fixed spacetime, the fundamental temporal order is absent. Consequently, the entire classical concept of causality, where a cause must happen before its effect, can no longer be fundamentally established. The distinction between a definitive 'Cause' and a definitive 'Effect' becomes untenable as Russell foresaw. When the separation between cause and effect becomes untenable, then the obvious question arises: why even preserve an untenable separation?

Therefore, the Effect Propagation Process framework adopts the causaloid, a uniform entity proposed by Hardy[9], that merges the 'cause' and 'effect' into a single entity. Instead of dealing with two nearly identical concepts discernible from each other by temporal order, the causaloid is a single entity that applies its "input" (cause), to a causal function that derives its "output" (effect), and whose relations to other causaloids define the causal structure and whose effect then propagates to the next causaloid. The nature of the causal function is not prescribed, allowing the Causaloid to encapsulate diverse logical forms, including but not limited to:

- A deterministic rule (IF temp > 100).
- A formal Structural Causal Model (SCM).
- A probabilistic estimate or Bayesian network.
- A specialized neural network.

For the deterministic case, the causal function takes some evidence as input, applies Boolean operators (AND, OR) or comparators, and returns a Boolean value as its PropagatingEffect. For a more complex causal scenario, the causal function encapsulates a set of structural causal equations, applies the corresponding calculus, and returns a probability distribution value as its PropagatingEffect.

In case of a probabilistic estimate or a Bayesian network, the causal function implements a Conditional Probability Table (CPT) or a similar probabilistic model, applies a probabilistic calculus, i.e., the chain rule of probability, and returns another probability as its PropagatingEffect. If a Causaloid receives multiple PropagatingEffects, each carrying a probability, the receiving Causaloid implements the aggregation of all probabilities. This is a deliberate architectural principle rooted in the EPP's primary role as a flexible, hybrid framework. A specialized neural network embedded into a causal function will most likely return a classification score as its PropagatingEffect. In case the output of the neural network results in a complex type, for example, generative data, then it is sensible to write its output into the appropriate context as a contextoid and return the context and contextoid ID as the PropagatingEffect.

It is important to note that the EPP framework adopts the conceptual role of the Causaloid as a spacetime-agnostic unit of causal interaction, inspired by Hardy's work on Quantum Gravity, but it does not use Hardy's formal definition that requires a complex process matrix. Instead, the EPP formulates the Causaloid as an abstract data structure that embeds a causal function, thereby decoupling it from any particular physical theory while preserving its core philosophical utility and making it practically implementable in software.

The term "propagation" refers to the fundamental process by which an effect is transferred within the structure from one Causaloid to another. This fundamental process is what gives rise to the appearance of propagation through spacetime in the classical view. Furthermore, while classical causality relies on a definite temporal order, the Effect Propagation Process treats temporal order as an emergent property, arising from the fundamental process itself.

While the Effect Propagation Process involves the transfer of effects within the fundamental structure, it is crucial to distinguish this from mere accidental correlation. The process reflects the fundamental way the underlying structure of reality establishes dependencies between its components and how it gives rise to the non-accidental relationships we recognize as observed causal relations. This fundamental determination, rather than simple co-occurrence, is what the "Effect Propagation Process" captures at the deepest level.

The Effect Propagation Process fundamentally inverts the notion of causal structure. Before the EPP, the causal structure was presupposed (i.e., DAG, a set of equations) in which variables, events, objects are placed. The consequence of this order is the fixed and often flat causal structure observed in classical methods of computational causality. The EPP, however, inverts the order and puts the causal entities first as monoidic primitives, and then introduces structure as relationships between these. This results in the operationalization of the fractal, self-referential causal structure that directly follows from the adopted single axiomatic generalized definition of causality.

Specifically, the self-referential structure materializes from the following:

- What is a causal structure? A set of relationships between Causaloids.

- What is a causaloid? An entity that defines its relationships to other Causaloids.

- What is a relationship? The pathway of propagating effects between Causaloids.

Therefore, a set of causaloids can represent arbitrarily complex causal relationships because of this fractal, self-referential definition. It also follows from the definition that causality in the EPP has three primary modalities:

- When the causal relations are fixed, the structure is static.

- When the causal relations are changing, the structure is dynamic.

- When the causal relations are brought into being, the structure emerges.

The core principle, that the entities define the structure and the structure is defined by its relationships between entities, holds true for all three modalities and, with it, establishes a principled foundation for dynamics. Because of its flexibility, the EPP can express static causal relationships similar to Pearl's Causal DAG, it can handle probabilistic causal systems similar in spirit to Dynamic Bayesian Networks, but then goes further and adds causal emergence and unifies both paradigms into one that is static and dynamic, deterministic and probabilistic while remaining structurally agnostic and thus allows for flexible geometric representation.

4.6   Causaloid Collection

Many real-world causal scenarios are not defined by a single cause but by the interplay of multiple factors. In practice, multiple modalities of causal aggregation may occur. For example, from a set of known causes, all of them must be true for an effect to occur. Then, in some cases, only one of many potential causes may lead to an effect. In other cases, more than one potential cause might be needed to trigger an effect, but not all known causes are required. To model these common collective structures, the EPP provides the Causaloid Collection.

A Causaloid Collection is a first-class entity that encapsulates a set of Causaloids and an explicit, configurable Aggregate Logic. This logic dictates how the individual PropagatingEffects of the member Causaloids are combined into a single, definitive outcome for the collection as a whole. This provides an ergonomic and principled way to express common causal patterns:

- **Conjunction (All):** The collection is active only if all of its member Causaloids are active.

- **Disjunction (Any):** The collection is active if at least one of its member Causaloids is active.

- **Absence (None):** The collection is active only if none of its member Causaloids are active.

- **Threshold (Some(k)):** The collection is active only if at least a specific number, k, of its n member Causaloids are active.

In a conjunction, the causal collection is active only if all of its member Causaloids are active. For a disjunction, it is sufficient when just one cause of the causal collection is active. Negation inverts the conjunction in the sense that it's only true when all causes in a causal collection evaluate to false. For a threshold-based logic, the causal collection is active only when at least some K out of the collection of n causes are true.

The causal collection might seem to break from the EPP's geometric foundation, as the relationships between its members are not defined by explicit hyperedges. However, the relationship of a Causaloid within a collection is not to

its peers, but to the collection's aggregate logic itself. For example, a collection with All logic is the formal equivalent of a single hyperedge connecting all member Causaloids as a source set to a single target representing a logical AND gate. Disjunctive and other logics follow a similar principle, allowing for a simpler and more intuitive alternative to modeling these common use cases with a full hypergraph. By providing the causal collection, the EPP allows for the concise and efficient modeling of the most frequent types of causal structures.

It is important to note that the configurable aggregate logic is only applicable to the causaloid collection. For a single causaloid, there is nothing to aggregate, and for the causal graph, the pathway through the graph provides the order of evaluation, therefore aggregation does not apply.

The Causaloid Collection proves particularly useful, for example, when modeling sensor fusion logic (k-of-n), multi-source object detection (All), or verifying safety interlocks (None). The verification of safety interlocks is a particularly good example for causal collections because it allows for clear, verifiable, and auditable encoding of standard safety protocols commonly found in regulated industries such as robotics and avionics. For use cases that require nested or arbitrarily complex causal relationships, the EPP provides the Causaloid Graph.

4.7   Causaloid Graph

Modeling real-world dynamic causal systems requires a mechanism capable of managing complexity. Classical computational causality relies on algebra, which is rich in formalization, but has its limits when complexity grows and thus limits scalability. The EPP adopts a geometric approach by expressing causal models as a hypergraph. The EPP exchanges the arithmetic complexity of solving large equation systems for the challenge of managing structural complexity that comes from the geometrization of causality.

In the EPP, the fractal, self-referential definition of causality directly translates to isomorphic recursive composition that enables concise expression of complex causal structures to manage structural complexity. A causal hypergraph may contain any number of nodes with any number of relations to other nodes, with each node representing a causaloid. A causaloid uniformly represents three distinct levels of abstraction:

- Singleton Causaloid: The base case, representing a single, indivisible causal mechanism.
- A Collection of Causaloids: A set of Causaloids that can be evaluated with an aggregate logic.
- A Causaloid Graph: A node can encapsulate an entire graph

Recursive isomorphism allows for building causal models in a modular and hierarchical fashion. A complex sub-system can be modeled as a self-contained Causaloid Graph, then encapsulated into a single node to be used as a component in a larger, higher-level model. The causal graph enables the concise expression of deeply layered systems without sacrificing logical integrity. The architecture of the Causaloid Graph is the direct physical manifestation of the EPP's core axioms. In accordance with the definition, the pathway of propagating effects, the relationship between Causaloids, is the hyperedge that connects the causaloid nodes. The Effect Propagation Process is the operational dynamic on this graph. When triggered, Causaloids are evaluated. Their outcomes, the PropagatingEffects, propagate along these hyperedges to other Causaloids, which in turn evaluate their own functions, thus continuing the process until the graph traversal completes and a final, reasoned inference is reached.

4.8   PropagatingEffect

The EPP emphasizes uniformity and, just like the causaloid folds cause and effect into one uniform unit, the PropagatingEffect folds causal input and output into one uniform unit. Its purpose is to distill the result of the node's reasoning into a clear, actionable directive that is passed to the next causaloid as its input. The PropagatingEffect represents a unified inference outcome across different reasoning modalities. By design, the EPP supports the following reasoning modalities:

- Deterministic
- Probabilistic
- Mixed, deterministic and probabilistic

The deterministic mode facilitates logical reasoning using Boolean logic. A state is either true or false. The EPP recognizes that, while there is a clear use case for deterministic causal reasoning, there is also an equally important use case for probabilistic causal reasoning. Also, the EPP only provides the reasoning mechanism, but leaves the exact details of the reasoning mechanism as implementation details of the causaloid. The wisdom of this decision comes from the realization that the EPP may be used in different scenarios with different requirements and therefore it leaves

the exact reasoning details to the practitioner. One important detail on the mixed reasoning. While it is designed internally to convert all Boolean state to probabilities and then reasons only over probabilities, its final outcome is actually a deterministic Boolean relative to a probabilistic threshold. In the DeepCausality implementation, all reasoning modes are traits with a default implementation thus leaving the practitioner the option to overwrite the default mode with a custom implementation when the need arises. Because the mixed modalities require different input and output types, the PropagatingEffect is isomorphic recursive to represent a variety of different reasoning types such as:

- Primitive Types: Deterministic, Numerical, and Probability values
- Complex Structures: Map or Graph for passing complex, structured, or relational data between causaloids.
- Contextual Link: A reference to a specific fact in the context to find the PropagatingEffect .
- None: Explicitly represent no effect.
- Referral: A type that contains a PropagatingEffect and reference to another causaloid to process it.

The Contextual Link accommodates for advanced causal reasoning via non-numerical representations by writing a complex reasoning outcome directly into the context and then propagating the reference to the next reasoning stage which reads the complex reasoning outcome and processes it further. For example, the Symbolic context type combined with the Contextual Link establishes a foundation for a uniform integration of multi-modal causal reasoning with advanced neuro-symbolic reasoning in a future revision of the EPP.

4.9 Dynamic Reasoning Modalities

The EPP uses context, the causaloid, and the propagating effect to enable several distinct modes of causal reasoning:

- **Static Reasoning**
- **Dynamic Reasoning**
- **Adaptive Reasoning**
- **Emergent Reasoning**

4.9.1 Static Reasoning

The entire causal graph is traversed according to its pre-defined structure. The reasoning is static because the reasoning path is always fixed. There is a large category of static causal models in practice that needs this modality. Furthermore, just by making the static context dynamic, a fundamentally static causal model becomes more valuable without any modification as the core dynamics is shifted to the context.

4.9.2 Dynamic Reasoning

A sub-graph or a specific path (such as the shortest path) is evaluated, allowing for targeted, context-dependent reasoning within the larger graph. It is dynamic reasoning because the pathway through the graph is determined at runtime via one of the EPP methods for graph traversal. Furthermore, the context of a dynamic reasoning graph might be either static or dynamic. In practice, usually a combination of both is used. An immutable static context for factual data assumed to remain invariant. A classic example would be the ICD-10 catalog. And then one or more dynamic contexts to model various data streams for real-time causal inference.

4.9.3 Adaptive Reasoning

The Causaloid itself determines the next step in the reasoning process conditional on its reasoning outcome. Based on its own internal logic, a Causaloid can dynamically dispatch the flow of causality to another Causaloid in the graph, enabling adaptive reasoning. To illustrate conditional reasoning, a clinical patient risk model may operate very differently for patients with normal blood pressure compared to high blood pressure patients. Therefore, two highly specialized models are defined and a dedicated dispatch causaloid queries its context for the latest blood pressure measurements and its recent history, uses its internal logic to decide the level of blood pressure and then dispatches all further reasoning to the specialized causal model for the detected blood pressure level. Adaptive Reasoning can be combined with either a static or dynamic causal graph. In a static graph, all potential reasoning pathways must be defined upfront. In a dynamic causal graph, new causaloids might be added dynamically, i.e., in response to a changing context and then the dispatch causaloid gets replaced with a new one that can dispatch to the newly created reasoning causaloids to enable dynamic adaptive contextual reasoning.

4.9.4   Emergent Reasoning

In case a dynamic context is combined with a dynamic causal reasoning graph that may even use adaptive reasoning, the fundamental assumption of causality, determinism, may not hold true any longer. Intuitively, when the reasoning graph is dynamically constructed and possibly mixed with adaptive reasoning, one would guess that determinism might not be feasible any longer. However, that is not necessarily the case. When the rules of dynamic graph modification are carefully crafted and verified, then even in this complex scenario determinism might hold true. However, the actual problem is the dynamic context in this case. When context is constructed dynamically in response to, say, a sensor data stream and relative to the detected shift in sensor data the reasoning graph is modified dynamically, then the entire premise of determinism hangs on the generative function of the context and that is exactly where formal verification falls apart. It is simply not possible to prove that a context, dynamically constructed from a sensor stream, can be deterministic and therefore the inter-dependent dynamic causal graph, regardless of its reasoning modality, cannot be proven to be deterministic either.

It is important to understand that the "unprovability" roots in the system's coupling with an open, unpredictable world facilitated via the dynamic context. The precise problem isn't the existence of a dynamic adaptive context, but the door it opens to an open, unpredictable world that may produce states that fall outside the boundaries of the causal model that interacts with the open world through the dynamic context.

4.9.5   Trade-off: Adaptability vs. Verifiability

There is a fundamental trade-off between adaptive and emergent reasoning in terms of adaptability and verifiability.

- **Adaptive Reasoning (Deterministic Dynamics)**
- **Emergence (Non-deterministic Dynamics)**

Dynamic and adaptive reasoning in the EPP is fundamentally deterministic, but limited in adaptability. These systems may grow complex, but fundamentally remain deterministic and therefore can be formally verified. Here, the trade-off is to preserve determinism at the expense of dynamic adaptability to enable a large class of problems where the modalities are reasonably well known upfront. For example, a clinical model that can switch between a "Normal Response" subgraph and a "Drug Resistance" subgraph is adaptive relative to the known spectrum of modalities. A financial model that switches between "Bull Market" logic and "Bear Market" logic is adaptive relative to these two market modalities. Because the set of possible causal models is closed, pre-defined, and pre-validated, the system's dynamic behavior remains fully deterministic and verifiable. Here, the EPP provides a formal architecture for building systems that can adapt to changing contexts without sacrificing safety and predictability.

Emergence on the other hand inverts the trade-off and enables dynamic adaptability at the expense of determinism and formal verification. Emergence is for systems where the causal rules themselves must be generated in response to dynamically emerging situations not foreseen by the designers. Even though determinism is not tenable any longer, restoring verifiability of dynamic emergence requires an entirely new set of methods.

The EPP deliberately supports both modalities to enable practitioners to choose the mode of dynamic that best fits their requirements and constraints. In regulated industries, emergence might be a non-starter, but a verified static causal model with a mixture of static and dynamic context might be able to get certified within clearly defined boundaries. In advanced research of system dynamics, emergence might be a good choice to explore and simulate scenarios that are otherwise complex to model.

In practice, it is realistic to see a largely deterministic system that leverages the full spectrum of static and dynamic context, static, dynamic, and even adaptive causal reasoning and only in very specific corner cases, a very small portion of a system would be designed as emergent albeit with carefully crafted boundaries to the generating functions. The EPP already enables a large part of dynamic use cases with dynamic context, dynamic and adaptive reasoning and therefore it is most likely that dynamic emergence will be reserved for a small and highly specialized set of use cases. In that sense, the EPP scales with growing requirements, from simple static models, complex dynamic models interacting with multiple contexts, to most advanced use cases of dynamic emergence.

4.10   Causal State Machine

The causaloid and causal graph provide the mechanism for causal inference, but they lack the ability of intervention. The EPP addresses this through the Causal State Machine (CSM), which serves as the formal bridge between causal reasoning and deterministic intervention.

The CSM originates in Finite State Machine (FSM) in that it aims to formalize state transition. However, a defining property of the Finite State Machine is its explicit 'Finiteness': the entire set of possible system states must be known at design time. The FSM paradigm is highly effective for closed-world problems where all conditions are predictable and known. However, the finiteness of states becomes untenable when applied to dynamic causality. The Causal State Machine generalizes the FSM and adapts it to operationalize interventions for dynamic causality through two mechanisms:

- A "Causal State" is an Inferred Predicate.
- The "Causal Action" is a Deterministic Intervention based on the Causal State.

In a classical FSM, a state is an identifier from a pre-defined list (e.g., "State A"). In the CSM, a Causal State is an inferential predicate defined as a specific Causaloid whose truthfulness is evaluated. The CSM does not need to know all possible states in advance. It only requires the causal logic (the Causaloids) necessary to infer whether the encoded predicate in the "Causal State" is true.

Each Causal State is formally linked to a Causal Action. This is a deterministic, programmatic function that is executed if and only if its corresponding Causal State is inferred to be true. This action represents a real-world intervention.

The CSM is an inference-to-action state machine that is both deterministic in its execution and dynamic in its definition. The CSM is deterministic within its encoded causal states and actions, but also dynamic in its definition as it can be extended at run-time by adding new Causal States and Actions, enabling the control logic of a system to evolve in tandem with the causal understanding of its environment.

4.11 Teloid and Effect Ethos

The EffectEthos acts as a deontic (rule-based) gatekeeper for the CSM. The CausalStateMachine is responsible for determining what could happen based on causal logic, while the EffectEthos determines what should happen based on a predefined set of rules and goals. This adds a layer of governance on top of the CSM and allows to enforce safety protocols, ethical guidelines, or business rules before executing actions.

A CSM is optionally configured with an EffectEthos upon creation. If an ethos is provided, it must be pre-verified, meaning its internal graph of norms is validated and ready for inference. The interaction begins when the CSM evaluates one of its CausalStates and a state becomes "active" if its internal evaluation yields true. This is where the EffectEthos comes into play. Before firing the CausalAction associated with the active state, the CSM performs the following checks:

- It verifies if an EffectEthos was configured. If not, it fires the action directly, and the process ends here.
- If an ethos exists, the CSM retrieves a Context from the active CausalState, which is crucial as it provides the situational background for the ethical evaluation.

With an active state, a context, and an ethos, the CSM then creates a ProposedAction. This is a temporary data structure that represents the action that is about to be taken. It then evaluates the ProposedAction using the provided Context. The EffectEthos then uses its internal graph of norms (Teloids) to infer a Verdict on the proposed action.

The EffectEthos performs the following steps for norm resolution:

- Filtering: It uses the provided tags to find a candidate set of relevant Teloids
- Activation: It executes the activation_predicate of each candidate Teloid against the Context.
- *Conflict Resolution*: It traverses its internal TeloidGraph to resolve conflicts among active norms using defeasible reasoning (e.g., Lex Specialis, Lex Posterior), producing a final, consistent set of justified norms before reaching a vedict.

The CSM inspects the Verdict returned by the ethos:

- Impermissible: If there is just one norm violated, the proposes action will be denied.
- Obligatory: The action must be taken and the causal action will be fired.
- Optional: In case an action is deemed optional, the CSM receives a cost estimate associated with the optional so it can decide if the action should be fired relative to a budget stored in the context.

The EffectEthos acts as a deontic gatekeeper for the CSM and it provides a mechanism to evaluate whether a causally-triggered action should be executed, based on a set of predefined norms, rules, or ethical principles.

4.12 Mapping of Classical Causal Models to the EPP

One key property of the generalized definition of causality used by the EPP is that classical causality can be expressed as a specialized case. In practical terms, that also means that methods build upon the classical definition of causality can be expressed through the EPP. The following subsections demonstrate that is indeed that case for Pearl's Ladder of Causation, Dynamic Bayesian Networks, Granger Causality, Rubin Causal Models, and Conditional Average Treatment Effects (CATE). For a full formalization of the subsumption of classical methods, see section 9.11 and for a formal proof that the EPP is a strict generalization of the SCM, see Appendix A.

4.12.1 Mapping Pearl's Ladder of Causation to the EPP

Judea Pearl's Ladder of Causation[4] defines three distinct levels of ability required for causal reasoning: Association, Intervention, and Counterfactuals. The EPP achieves these three rungs of the ladder by different means than the established methods of the SCM and Causal DAG.

Rung 1: Association

The first rung, Association, concerns reasoning from observational data. It answers the question, "What is the likelihood of Y, given that we have observed X?" In classical models, this is handled by conditional probabilities, i.e., $P(Y|X)$.

In the EPP, association is a structured, operational process:

1. The observation (X) is formalized as Evidence and presented as an input to a Causaloid.
2. The background condition $(|)$ is represented by the context, which provides the necessary supporting data for the reasoning process.
3. The causal inference of (Y) is performed by the causal function embedded within each Causaloid. This function takes the Evidence and any required information from the Context as its inputs and computes a result.
4. The inference result is emitted as a PropagatingEffect, which then travels along the graph's hyperedges, serving as Evidence for subsequent Causaloids.

Thus, "seeing" in the EPP is the operational dynamic where initial Evidence triggers a cascade of computations via the causal functions throughout the Causaloid Graph, leading to a final, reasoned inference.

Rung 2: Intervention

The second rung, Intervention, involves predicting the effects of deliberate actions. It answers the question, "What would Y be if we do X?" This is formalized in Pearl's framework by the do-operator, which simulates an intervention on the causal model itself.

The EPP provides a mechanism of intervention through the Causal State Machine (CSM). The CSM links causal inferences to deterministic actions:

- A Causal State is defined as an inferential predicate. It is a specific Causaloid evaluated against a specific Context.
- This Causal State is mapped to a Causal Action, a verifiable function that executes when its corresponding state is inferred to be true.

This Causal Action is the EPP's intervention. It is a programmatic function that changes state. This change is then reflected as an update to the context, creating a complete feedback loop of inference, action, and new observation.

Rung 3: Counterfactuals

The third rung, Counterfactuals, involves reasoning about alternative possibilities given a known outcome. It answers the retrospective question, "What would Y have been if X had been different, given that we actually observed Z?"

The EPP's architecture provides a Contextual Counterfactual mechanism, which leverages the EPP's externalization of context:

1. Abduction is Context Pinning
2. Action is Contextual Alternation
3. Prediction is Re-evaluation over an altered context.

The factual observation (Z) is already explicitly represented within the primary context. The abduction step is therefore equivalent to identifying and "pinning" this factual context.

The hypothetical premise ("if X had been different") is then established by creating a new, hypothetical context $(C_{counterfactual})$ by cloning the primary context $C_{factual}$ and modifying the value of the relevant Contextoid. The system then executes the exact same, unmodified Causaloid Graph, but uses $C_{counterfactual}$ as its frame of reference. The resulting inference is the answer to the counterfactual query.

The EPP's mechanisms of causation differ from Structural Causal Models, but they fulfill the same fundamental goals of 'seeing,' 'doing,' and 'imagining' Judea Pearl established via the ladder of causation. Table 1 summarizes the comparison of the EPP to the existing methods of computational causality.

Table 1: Comparison of Causal Ladder Implementations

Ladder Rung	Pearl's Framework (SCM/DAG)	Effect Propagation Process (EPP)
1. Association	Statistical analysis; calculating conditional probability $P(Y|X)$.	Execution of a `causal function` on `Evidence` within a factual `Context` (C_{factual}).
2. Intervention	The *do*-operator; surgical modification of the model's structural equations.	Execution of a `Causal Action` by the `Causal State Machine (CSM)` in response to an inferred state.
3. Counterfactuals	Three-step algorithm: Abduction (solving for latent variables), Action (model surgery), and Prediction.	Three-step process: Context Pinning, Contextual Alternation, and Re-evaluation of the *unmodified model* over an *altered context* $(C_{\text{counterfactual}})$.

The decision to separate causal logic (the Causaloid Graph) from its data (the Context) that underpins contextual alternation leads to some welcome properties. For example, through contextual alternation, counterfactual reasoning becomes an "embarrassingly parallel" problem because, if 100 alternate contexts are derived, all of them are independent from each other and thus can be evaluated in parallel.

4.12.2 Mapping Dynamic Bayesian Networks to the EPP

Dynamic Bayesian Network is the established framework for modeling dynamic causal systems. A DBN models a temporal process by "unrolling" a causal graph over discrete time slices, creating separate nodes for a variable at each point in time. The EPP represents this same process by evaluating a single, static causal model over a dynamic, temporal context hypergraph. The mapping of DBN to EPP components is as follows:

Time Modeling:

In a DBN, time is an implicit index of the temporal variables. In the EPP, time is made explicit within a context by representing each time slice (e.g., X_{t-1}, X_t, X_{t+1}) as a Tempoid, a temporal type of Contextoid, with their sequential relationship defined by hyper-edges within the context hypergraph. In the EPP data might be attached to a Tempoid as a dedicated node of a different type, i.e., a Datoid.

State Variables:

A state variable in a DBN (e.g., the concept of Weather across time) corresponds to a single Causaloid in the Causaloid Graph. The Causaloid represents the variable's underlying causal mechanism and the state of "Weather at time t" is the result of evaluating the "Weather" Causaloid contextually using the Tempoid that maps to the time t. A contextual temporal graph allows a Causaloid to access uniformly multiple slices of time at different scales, i.e., weekly average rainfall and today's rainfall to inform its causal logic.

Dependencies:

The directed edges in a DBN can reference within the same time slice or across different time slices. For edges within the same time slice, (e.g., Weather$_t$ → Umbrella$_t$), the causal function simply references its causal logic to the one tempoid at time t in the graph. If "Weather" is a complex state object, then the causaloid may loads it from another context before evaluating the causal rule that would lead to Umbrella become true.

For edges between different time slices (e.g., Weather$_{t-1}$ → Weather$_t$) are represented by the hyperedges in the Causaloid Graph by referencing two different Tempoid nodes. Practically, one would implement a dynamic temporal graph index with accessors for frequently used "current" or "previous" values.

Conditional Probability Tables (CPT):

Conditional Probability Tables (CPTs): The CPT defining a variable's probability given its parents (e.g., $P(X_t \mid Z_t, X_{t-1})$) is implemented as the causal function within the Causaloid.

Execution Flow:

To compute the state of the system at time t, the causal function of a Causaloid queries the Context to determine the current time slice. It receives the PropagatingEffects from its parent Causaloids (representing their states at the appropriate time slices) and uses its internal CPT logic to compute a new probability distribution. This distribution is then emitted as a new probabilistic PropagatingEffect. This process maps the EPP to the "filtering" or "unrolling" inference of a DBN.

4.12.3 Mapping Granger Causality to the EPP

Granger causality is a foundational concept for causal inference in time-series data, particularly in econometrics. It is used to solve practical problems, for instance, determining if past changes in the price of oil can be said to "Granger-cause" future changes in shipping industry activity. A time series $X(oilprice)$ is said to "Granger-cause" another time series $Y(shippingactivity)$ if the past values of X contain information that helps predict the future values of Y better than using the past values of Y alone. It is fundamentally a test of predictive utility. The EPP's architecture, with its first-class treatment of time and context, provides a natural and powerful framework for modeling this scenario:

- **Time-Series as a Temporal Context:** The historical data for both oil prices and shipping activity are represented within the EPP context. This is modeled as a sequence of Tempoid nodes (representing months or quarters), each linked to Datoid nodes containing the respective values for $X(oilprice)$ and $Y(shippingactivity)$ at time t.

- **The Predictive Model as a Causaloid:** The predictive model for shipping activity is encapsulated within a dedicated 'Causaloid'. This Causaloid's purpose is to predict the next value of shipping activity, Y_{t+1}, by querying the 'Context' for past values.

- **The Granger Test as a Counterfactual Query:** The core question of "Do past oil prices improve the prediction of future shipping activity?" is a fundamentally counterfactual query. The EPP models this directly using its Contextual Alternation mechanism:

  1. A "Granger Test" Causaloid initiates two parallel, hypothetical evaluations.
  2. In the first evaluation, the Causaloid is allowed to query the complete, factual 'Context', which contains the history of both oil prices and past shipping activity.
  3. In the second evaluation, the Causaloid is evaluated against a hypothetical, alternate 'Context', C', where the history of oil prices is deliberately excluded or masked.
  4. The "Granger Test" Causaloid then compares the prediction error from both evaluations. If the error is significantly lower in the evaluation that included the history of oil prices, the test concludes that oil price Granger-causes shipping activity.

The EPP's temporal Context is not restricted to linear time steps; it can be a rich hypergraph representing multiple time scales (e.g., daily price volatility and quarterly shipping trends). Furthermore, the Causaloid is not restricted to the linear models typical in econometrics. This allows for the construction of more sophisticated, non-linear, and multi-scale versions of Granger-causal tests

4.12.4 Mapping Rubin Causal Model (RCM) to the EPP

The Rubin Causal Model (RCM), also known as the Potential Outcomes framework, is a cornerstone of modern causal inference, particularly in statistics, econometrics, and the social sciences. The RCM defines the causal effect on a specific unit, i, as the difference between two potential outcomes: $Y_i(1)$, the outcome if the unit receives the treatment, and $Y_i(0)$, the outcome if the unit does not receive the treatment. The central challenge of the RCM, termed the "fundamental problem of causal inference," is that for any given unit, only one of these two potential outcomes can ever be factually observed.

The EPP models the scenario of potential outcomes as contextual alternations. From a base context, say C_i, two alternate contexts are derived, one C'_i with the treatment applied and another one, C''_i with the control applied. From there, the two potential outcomes for unit i are therefore defined as:

- $Y_i(1)$ is the final 'PropagatingEffect' that results from evaluating the system's 'CausaloidGraph' against a context, C'_i where the treatment has been applied.

- $Y_i(0)$ is the final 'PropagatingEffect' that results from evaluating the exact same 'CausaloidGraph' against another hypothetical context, C_i'' where the control has been applied.

- Estimate the difference between the PropagatingEffect resulting from $Y_i(1)$ and the PropagatingEffect resulting $Y_i(0)$ as the causal effect on a unit i.

From the EPP's perspective, the "fundamental problem of causal inference" is a physical constraint that only applies to physical measurements. The EPP, as a computational framework, treats counterfactuals via its contextual alternation mechanism. The EPP can instantiate and evaluate the causal outcomes for both the treatment and control contexts in parallel, thus estimating the difference between two potential outcomes. This mapping demonstrates that the RCM's core concepts are a natural fit within the EPP's architecture and the EPP provides a mechanism to extend the RCM with a richer, dynamic, and non-Euclidean context.

4.12.5 Mapping Conditional Average Treatment Effects (CATE) Inference to the EPP

The estimation of Conditional Average Treatment Effects (CATE) is a critical goal of modern applied causal inference, particularly in domains like personalized medicine. CATE is defined as the expected causal effect of an intervention for a specific individual or sub-population, conditioned on their unique characteristics. The purpose of CATE is to move beyond population averages and determine for whom a treatment is beneficial, harmful, or ineffective.

It achieves this by leveraging the Rubin Causal Model (RCM). CATE is formally expressed as $\tau(x) = E[Y(1) - Y(0)|X = x]$, where $Y(1)$ and $Y(0)$ are the potential outcomes. While the definition is elegant, applying it to real-world observational data is a challenge, as only one potential outcome is ever observed. This challenge necessitates a two-stage process:

- Stage 1 (Discovery): Data science discovers and validates causal links.
- Stage 2 (Inference): The causal links are operationalized to answer specific CATE queries.

It is crucial to position the Effect Propagation Process correctly within the broader landscape of computational causality. The EPP is a foundational framework for causal reasoning, simulation, and operations. In its current form, it is not a tool for automated causal discovery from raw observational data. For contextual causal learning within the EPP, more work will be necessary. Therefore, all existing methods of causal discovery remain indispensable for inferring causal graphs from observational data. In an EPP-based workflow, these tools are invaluable for building and validating the initial causal graph. A function learned via a tool like DoWhy or EconML can be directly encapsulated within a Causaloid. The following mapping of CATE inference presumes a causal graph has been found and validated with existing methods and thus is scoped to operationalize the inference stage of CATE. The inference stage of CATE maps to the EPP in two different modalities: Static and Dynamic.

Representing Static CATE in the EPP

The EPP models a static CATE query as a formal, computable, counterfactual simulation. This is achieved by mapping the core concepts of the query onto the EPP's architectural primitives:

- **The Condition $X = x$ as a Static Context:** The unit's complete set of pre-treatment covariates, $X = x$, is represented by a static `Context` (C_x). This context serves as the factual "ground truth" for the unit's state at the moment of the query.

- **The Learned Function as a 'Causaloid':** The causal relationships and the CATE function ('$\tau(x)$') that were discovered in Stage 1 (e.g., using DoWhy) are encapsulated within the causal logic of one or more `Causaloids`. For a simple case, a single 'CATEstimator' `Causaloid` might contain the entire learned function. In a more complex model, this might be a full `CausaloidGraph` representing the mechanistic pathways of the treatment.

- **Potential Outcomes via 'Contextual Alternation':** The EPP computes the potential outcomes, '$Y(1)$' and '$Y(0)$', not through statistical adjustment, but through direct, parallel simulation. This is achieved via the EPP's core counterfactual mechanism:
  1. **Abduction:** The factual, pre-treatment state of the unit is established by "pinning" the static context, 'C_x'.
  2. **Action:** The system then creates two hypothetical, parallel realities by cloning this base context:
     - A 'treatment' context ('C_1') is created where the treatment is applied (e.g., by introducing specific `Evidence` to a "treatment" `Causaloid`).

- A 'control' context ('C_0') is created where the control is applied.

3. **Prediction:** The EPP evaluates the entire, unmodified `CausaloidGraph` twice—once against the treatment context 'C_1' to compute '$Y(1)$', and once against the control context 'C_0' to compute '$Y(0)$'.

- **The CATE as the Final 'PropagatingEffect':** The final CATE value, '$\tau(x)$', is the directly computed difference between the two 'PropagatingEffects' representing the potential outcomes. The entire workflow can be encapsulated in a single 'CATECausaloid' that orchestrates this simulation and returns the final estimate.

4.13 Comparison

The established methods of computational causality, particularly the frameworks developed by Pearl, Rubin, and their successors like Bareinboim, represent monumental intellectual achievements that form the foundation of the field. These methods provide a powerful and mathematically rigorous toolkit to infer a fixed causal structure (the DAG) and reason within it, even with missing data and latent confounders. At its core, the existing methods enable causal reasoning in a world of stable systems and static laws.

The Effect Propagation Process (EPP) framework re-architects causality from first principles as an inherently dynamic, process-oriented, spacetime-independent functional dependency. It enables causal reasoning for dynamic systems in which the causal laws themselves can evolve. These systems exist, for instance, in quantitative finance that consistently deals with dynamic regime shifts, and system biology where dynamic pathways commonly occur. Table 2 provides a detailed comparison of the core philosophical and architectural differences between the existing methods of causality and the EPP:

Table 2: Comparison Between Classical Methods and the EPP

Feature	Classical Causality	Effect Propagation Process
Core Axiom	Causal Markov Condition & Faithfulness	Functional Dependency
Causal Structure	Static	Dynamic & Emergent
Role of Spacetime	Implicit Background	Explicit & Computable Context.
Representation	Euclidean-centric	Natively Non-Euclidean
Fundamental Unit	Variables	The Causaloid
Primary Goal	Causal Discovery & Inference	Dynamic Causal Modeling

From table 2, it is evident that classical methods operate within a defined static structure, whereas the EPP is designed for less certain, dynamic structures. Therefore, classical causality provides verifiability through determinism: given a static model, one can prove the outcome of an intervention. The EPP treats static models as a specialized case for which determinism still holds. However, for its dynamic modalities, the EPP trades the predictability of a fixed structure for greater expressive power, which necessitates a different method of verifiability. The EPP provides this by shifting the focus of verification from system states to system intent. It introduces a computable teleology, enabling the formal verification of a system's actions against a predefined purpose (telos) and ethical constraints (ethos). Thus, while a system's causal structure may emerge, its behavior can be proven to remain aligned with its core principles as a prerequisite for establishing verifiable safety in complex adaptive systems.

4.14 Selection Criteria

In cases where methods of classical causality and conventional machine learning do not solve the problem at hand, methodologies rooted in EPP might be preferred. Table 3 provides guidance on selecting an appropriate causal modeling approach based on the temporal and spatial complexity of the system under investigation.

Emergent Causality: If the problem involves emergent causal structures (where the causal graph itself is not fixed and changes dynamically based on context, i.e., dynamic regime shifts), EPP-based methodologies become the only viable option.

Feature Assessed	System Characteristic	Classical Methods	EPP Methods
Temporal Complexity			
	Single timescale + linear progression	Preferred	
	Multiple timescales OR nonlinear temporal relationships	May struggle	Consider EPP
	Multiple timescales AND nonlinear temporal relationships	Not Possible	EPP Preferred
Spatial Structure			
	Euclidean space with fixed coordinates	Preferred	
	Non-Euclidean OR dynamic spatial relationships	Limited/Difficult	EPP Advantageous
	Non-Euclidean AND dynamic spatial relationships	Not Possible	EPP Preferred
Combined Complexity			
Scenario 1	Linear Time & Euclidean Space	Preferred	
Scenario 2	Nonlinear Time OR Complex Spatial	Limited/Difficult	Consider EPP
Scenario 3	Nonlinear Time AND Complex Spatial	Not Possible	EPP Preferred

Table 3: Causal Method Selection Matrix

4.15 Discussion

The preceding sections have laid out the architectural components of the Effect Propagation Process. However, the capacity of the EPP to model dynamic systems where the causal structure itself can evolve introduces a class of challenges that reach beyond the scope of formalism alone. There is an important distinction between the reasoning modalities the EPP introduces:

- Static
- Dynamic
- Adaptive
- Emergent

For static, dynamic, and even adaptive reasoning determinism remains preserved. However, when a dynamic context is combined with a dynamic causal model, the resulting co-evolving of context and causal logic might not be deterministic any longer, and thus amounts to dynamic causal emergence. In practice, dynamic and adaptive causal reasoning already cover the vast majority of practical problems the EPP is designed to address. However, in situations where the dynamic change cannot be known upfront, the emergent modality closes the gap albeit at the expense of determinism which imposes a profound challenge. To ensure that the EPP is built upon a sound, coherent, and trustworthy foundation, it is required to first understand the inherent higher-order consequences of its dynamic design from first principles.

Therefore, the subsequent philosophical chapters establish the first principles of the EPP. The Metaphysics in section 5 will establish the EPP's first principles of being and change. The Epistemology in section 6 will explore the far-reaching consequences for knowledge and truth in such a system. The effect ethos and teloid are expanded in section 7. Finally, the Ontology in section 8 will use these insights to structure a safe and sound set of primitives that are ready for rigorous formalization in section 9 and implementation in section 10.

5 The Metaphysics of the Effect Propagation Process

The metaphysics of the Effect Propagation Process establishes a set of underlying principles widely used across the EPP. The three core metaphysical concepts are defined as: Monoidic Primitives, Isomorphic Recursive Composition, and Contextual Relativity. Then we derive from these first principles the metaphysics of being, dynamics, becoming, and doing. Combined, these form the foundation of orthogonal design used throughout the EPP and its implementation DeepCausality. Ontological design is a constructive, engineering-oriented form of philosophy to specify the necessary and sufficient conditions of a new system to exist and operate coherently. The EPP's metaphysics is, therefore, the blueprint for a computable reality. It establishes the axiomatic foundation upon which the ontology is built, the epistemology is derived, and the implementation is realized.

5.1 Monoidic Primitives

A monoid is defined as an abstract algebraic structure that comprises:

1. A set of elements with a certain type.
2. A binary operation that combines any two elements of the set results in a third element of the same set.
3. An identity element, which, when combined with any other element, leaves it unchanged.

Monoidic elements that can be combined with each other are fundamental to the composability of the EPP.

5.2 Isomorphic Recursive Composition

Isomorphism means "having the same shape" in the sense that isomorphic elements all share the same form. Recursion refers to a structure containing itself. Isomorphism enables the combination of different types of monoidic primitives whereas recursion allows self-referential nesting. Combining these two concepts results in isomorphic recursive composition. This composition enables a monoidic primitive to derive its significance from its structural relation to other monoidic primitives. Critically, to ensure the non-reducibility of form of monoidic primitives within the isomorphic recursive composition, it must have a singleton representation of itself.

5.3 The Metaphysics of Context

The metaphysics of context is structured as a classical "matter", the hyle, through the monoidic primitive of the contextoid. The contextoid entails one unit of context, that can be space, time, spacetime, symbol, or data. The context is free of recursion to prevent paradoxical states such as time-loops. Furthermore, the metaphysics does not prescribe any particular shape or structure of the context to ensure a context can be represented in a structure best suited for the domains at hand.

5.4 The Metaphysics of Being

The metaphysics of being is structured in a classical Aristotelian notion (Book Zeta[117]):

- Monoidic Primitives (Matter): The EPP is composed of fundamental, identifiable elements.
- Isomorphic Recursive Composition (Form): Gives the primitive elements its form.

Combined, the monoidic primitives and the isomorphic recursive composition form an Aristotelian hylomorphic compound. The 'Monoidic Primitives' constitute the "matter" (hyle) and the 'Isomorphic Recursive Composition' provides the "form" (morphe) that arranges its elements into a structured, meaningful whole.

This hylomorphic compound is the *archê kai aitia* of the system's existence. The specific form imposed upon the primitives is the foundational principle and explanation for its properties. Therefore, the hylomorphic compound is by definition static and describes what the system is at a snapshot in time, but it contains no inherent mechanism of change.

5.5   The Metaphysics of Dynamics

The classical hylomorphism describes a static substance as the combination of its matter and form. Therefore, to account for the dynamism inherent in the EPP, the *archê kai aitia* needs to capture the dynamics within an existing substance. However, because of the spacetime agnostic design of the EPP, dynamics can only be defined relative to its engulfing context. Therefore, the principle of contextual relativity operationalizes the expression of Substantial-Structural co-determination relative to its context. This is an inherent principle of how a substance with a fixed identity can exhibit variable states. By its definition, dynamics is predictable and bound to an existing substance and context.

The dynamics in the EPP operates relative to its engulfing context and within its structure. The structure might be static or dynamic, but for dynamic the separation between hyle (context), logos (logic) and its structure remains intact. That means, a causal logic can only operate along pathways. Dynamics cannot leave a pathway, it must follow it regardless of whether these paths were established upfront or created dynamically

5.6   The Metaphysics of Adaptation

For Adaptive dynamics as in adaptive reasoning, there is an element of interaction between the logos and its structure. The logos, a causaloid, can alter adaptively the pathway through the causal structure depending on its internal reasoning. Specifically, when a causal model is structured in such a way that it contains sub-models for specific scenarios, then a causaloid in adaptive reasoning mode can determine dynamically which sub-model is more applicable and then dispatch the reasoning path to the sub-model and with that alter the causal reasoning pathway. Notice, as long as the targets of the dispatch exist and the dispatch logic is decidable, the modality of adaptive dynamics remains deterministic. Although adaptive dynamics can change the reasoning pathway, however, it cannot fundamentally change or evolve the structure itself.

5.7   The Metaphysics of Structural Change

The principle of contextual relativity (Dynamics) accounts for how a system can change its state predictably within a fixed structure and enables reasoning along a fixed path. Adaptive reasoning extends the modality towards a dynamic pathway through a static or dynamic structure. However, none of it can account for how the structure itself might evolve, as is required in advanced use cases like autonomous navigation. For structural change, the principle of emergence allows for two modalities.

- Fixed structural change
- Dynamic structural change

Fixed structural changes limit systems to pre-defined all possible structural changes that are triggered by known conditions. Consequently, the system remains fully deterministic and verifiable, but at the expense of adaptability. Conversely, dynamic structural changes enable adaptability by generating novel causal structures that were not explicitly encoded before, but at the loss of determinism and verifiability. This fundamental trade-off determinism at the expense of adaptability versus adaptability at the expense of determinism demands a fundamental decision. Also, fixed structural change can be expressed as a specialized form of dynamic structural change with a momentum of change held constant. The reverse, however, is not possible.

The Effect Propagation Process as a theory of dynamic causality deliberately embraces dynamic structural changes. The limitations of fixed structural changes are too restrictive for the next generation of intelligent systems. The consequences of defining and implementing dynamic structural changes, while substantial, are accepted as the price of enabling truly dynamic causality in a dynamic environment. Therefore, the metaphysics of becoming defines the required principle of emergence, the subsequent ontology structures the mechanism of emergence, and the epistemology captures the implications. To capture dynamic structural change, the principle of Higher-Order Emergence becomes necessary.

5.8   The Metaphysics of Becoming

The principle of Higher-Order Emergence captures the profound process of creation of new substance from within an existing substance. Emergence brings into being a new matter, a new form, both of them, or new dynamics. The principle of Higher-Order Emergence operationalizes two different modalities.

First-Order Emergence explains the creation of new substance. While contextual relativity applies to an existing substance, it cannot bring into being a new substance. The principle of 'First-Order Emergence' posits a generative

capacity within the EPP, a capability of imposing a novel *archê kai aitia* upon a set of primitives. This is the mechanism by which a new substance from within an existing context comes into being.

In this modality, the distinction between the system and its context begins to dissolve. The result is a non-deterministic co-evolution where the spectrum of subsequent causal structures and contextual facts cannot be predicted any longer. The "reason for being" *aitia* of any given state is no longer a fixed principle but is itself an emergent property of the ongoing, self-modifying process. However, the process of becoming is invariant because of the first-order designation.

Higher-Order Emergence moves beyond the creation of a new substance from within an existing one. Instead, it describes a state where the generative capacity that enables 'Emergence' acts upon itself recursively in what amounts to a dynamic co-mergence of form, matter, and dynamics through recursive higher order emergence.

In this modality, the process of becoming itself becomes dynamic. This higher order emergence represents the EPP's most advanced state of becoming, one that necessitates a new epistemology of emergence to explore its inherent emergent properties.

Metaphysically, the principle of Higher-Order Emergence demands further elaboration. Fundamentally, it implies that first-order emergence, the capacity to create new substance, is simply a less recursive specialization of the governing higher order principle.The principle of Higher-Order Emergence leads to outcomes that are not necessarily guaranteed to be decidable let alone deterministic any longer and therefore it foreshadows three crises:

1. The Crisis of Justification

2. The Crisis of Truth

3. The Crisis of Explainability

The crisis of justification immediately results from the fact that it must be decided how to choose one state of emergence over another and how. The justification must be there otherwise the decision cannot be made, but this would render higher-order emergence fundamentally undecidable.

The crisis of truth results from the fact that, if emergence generates a new context, then how do we know that the facts in the newly generated context are true? If emergence generates new causal rules that uses new facts from a generated context, how do we know the outcome is true? If facts are fluid, verification is impossible. Therefore, truth must be re-established otherwise it undermines trust in operational safety of the EPP.

The crisis of explainability means that in co-emergence, it might not be possible any longer to explain the outcome because of the previous crises of truth and the crisis of justification. Furthermore, if the process of emergence itself cannot be explained, how could possible derived artifacts be explained? How can a system be held accountable when it's not explainable. It is not possible, and therefore the crisis of explainability roots in the very core of higher-order emergence.

The introduction of higher order emergence also raises the question of the genesis process, the origin of the emergence itself. Fundamentally, the genesis process imposes a decision: Do we allow any kind of machine intelligence to modify its genesis process or not? The author argues for an unequivocal no. Considering the alternative, when a system that can evolve its own genesis process, it would fundamentally become uncontrollable and unexplainable.

Therefore, the genesis process of emergence has to remain at the sole discretion of a human designer to ensure its explainability and a fundamental alignment with human values. The metaphysics itself cannot establish the core ethos or telos, only a human designer can do that. Under no circumstances should the genesis process in parts or in its entirety ever be created or modified by a machine intelligence because it cannot possibly have the innate ethos of a human being and thus cannot possibly align itself with humanity.

The existence of the genesis process also raises the thorny issue of whose ethics and values to codify and why? Which human designer? Who guards the guardians of the genesis? How to balance conflicts between different stakeholders? These are immense normative and political challenges and, a metaphysics alone, cannot possibly answer these fundamentally societal questions.

As a consequence, the genesis process itself is a decision with far reaching higher order effects. The mitigation of inevitable unintended higher order consequences, will lead to the necessity of an immutable genesis telos, an underlying intent that serves as a criterion to discern whether emerging states are intended.

5.9 The Metaphysics of Axiology

The reasoning (logos) alone does not have any impact without action, but action without reason (logos) may not lead to a desired outcome. The link between reasoning and action is captured in axiology, the study of value, ethics, intent, and aesthetics. For the EPP, two aspects of axiology are used.

The Telos, intent, which is the specific, singular intent or overarching goal applied proactively and the ethos, a framework of rules retrospectively applied. The goal of the telos is to prevent incorrect causal reasoning within a causaloid, for example, resulting from invalid context reading i.e. because of a faulty sensor.

The ethos with its finer set of rules is applied after all reasoning has been completed, but before an action can be taken. A well known property of complex systems is that, even when all components operate within a valid range, the final outcome can potentially still be invalid because of unforeseen state combinations. Therefore, the goal of the ethos is to prevent unforeseen consequences from happening by vetoing actions that would violate the rules codified in it.

In the EPP, the causaloid folds cause and effect into one single monoidic entity and the PropagatingEffect folds input and output into one single monoidic entity. In the same way, the teloid folds intent and ethos into one single monoidic entity with a twofold function:

Teloid: A single teloid codifies a single intent or a single rule. A causaloid can query one or more teloids to inform whether its reasoning stays within codified intent to ensure alignment.

Effect Ethos: Multiple teloids, using isomorphic recursion, form a scalable rule set that codify the effect ethos governing the final PropagatingEffect. Before a reasoning outcome stages into an action, it must pass the governing effect ethos to ensure that the proposed action is within the overall applicable rule set. Only after the checks enforced by the effect ethos have passed, a proposed action can be executed.

Together, the teloid and the effect ethos safeguard operations of the EPP regardless of the modality of dynamics. Furthermore, the three crises caused by emergence are mitigated:

1. The Effect Ethos justifies why an action was taken.
2. The Teloid codifies the intent and thus attests to the ground truth.
3. Both, the Teloid and Effect Ethos explain a system even when it is emergent.

Because dynamics and emergence are first class principles in the EPP, so are the teloid and effect ethos to safely govern the EPP and its actions.

5.10 The Metaphysics of Doing

In the EPP, the transition from insight to action is governed by a causal state machine that is based on two principles.

First, the causal state that takes as input any terminal PropagatingEffect and decides deterministically whether to act. The separation between reasoning outcome and deciding whether to act upon the reasoning outcome allows for arbitrary complex decision logic especially if the terminal PropagatingEffect is, for example, symbolic or probabilistic. In this causal state, the effect ethos intercepts the line of reasoning and decides whether the final determination to act falls within its rules.

Second, the causal action encodes what action to take conditional on the causal state. The causal action is an arbitrary and unconstrained function to allow for the broadest possible set of action an EPP system can take.

Combined, the causal state and action form the causal state machine governed by the effect ethos to safely transition from insight to action while respecting all rules encoded in the accompanying ethos.

5.11 Discussion

The complete metaphysics of the EPP is the synthesis of three core principles:

1. Context: The hyle, the eternalized context.
2. Substance (Being): The hylomorphic compound of Primitives and Composition.
3. Dynamics (Changing): Governed by the principle of Contextual Relativity.
4. Adaptive (Adapting): Governed by the principle of adaptive reasoning.
5. Emergence (Becoming): Governed by the principle of Higher-Order Emergence.

6. Effect Ethos (Judging): Governed by its axiology.

7. Acting (Doing): Governed by the effect ethos.

The EPP's core metaphysical principles, Monoidic Primitives, Isomorphic Recursive Composition combined with the principles of and Contextual Relativity and Higher-Order Emergence form the foundation of the EPP. The EPP provides a framework that distinguishes between two tiers of emergence. First-Order Emergence describes the system's capacity to generate new substance from a stable set of generative rules. Higher-Order Emergence, describes the system's ultimate capacity to evolve itself via a recursive and open-ended process of generative emergence. The introduction of higher-order emergence implies that the ultimate outcome of the emergent process is not guaranteed to be decidable let alone deterministic any longer. This is a deliberate design decision to provide multiple modalities for different requirements. For dynamic systems, contextual relativistic dynamics should suffice. For handling regime change where the new structure can be decided a-priori, first-order emergence should suffice. However, when handling dynamic relativistic regime change in response to an evolving context, higher-order emergence becomes necessary to capture the dynamic co-emergence, but it removes the deterministic foundation of causality, which introduces three foundational crises. The introduction of the teloid and effect ethos address these crises by establishing a priori and a posteriori policy check to ensure safe, reliable, and explainable operations. The EPP, through the introduction of Higher-Order Emergence and the effect ethos, establishes a foundation for exploring the interrelation between emergent systems and human value.

6 The Epistemology of the Effect Propagation Process

Epistemological approaches to acquiring knowledge in research fall into three categories: positivism, interpretivism, and pragmatism. Positivism concerns itself with observable facts based on the scientific method and thus seeks to achieve generalizability and objectivity. Interpretivism maintains that our knowledge depends greatly on our interpretation of observations of human actions, experiences, and environments, thus making interpretive research more subjective. Pragmatism focuses on practical effects or solutions to address problems that are suitable for existing situations or conditions. The epistemology of pragmatism is that knowledge is a self-correcting process based on experience; thus, it must be evaluated and revised in view of subsequent experience.

6.1 Overview

The presented EPP epistemology changes depending on whether the context is static or dynamic, and, equally profound, whether the EPP is static, dynamic, or emergent.

6.1.1 Ontology of Knowledge Sources

In the EPP, the context is designed as the source of factual knowledge. For context, facts may remain invariant (e.g., the value of Pi) or receive continuous updates. The designation whether a context is static or dynamic refers to its structure, not to the factual data in it. Furthermore, a context might be shared between two or more defined EPPs and an EPP may use one or more context(s), thus simplifying modeling complex domains. The EPP encodes each causal relationship in a designated Causaloid. The Causaloid encodes the causal rule, whereas the context encodes supporting data required to apply the rules. The Causaloid may use external data or data from the context to apply its rule. For example, a context may encode a continuous signal feed from a LIDAR sensor and the Causaloid encodes a rule to infer whether an obstacle has been detected. In this case, the context provides all data. In another scenario, a context may encode several known defect patterns, a Causaloid tests incoming image data for the defect data from the context, but uses incoming real-time image feeds from a manufacturing monitoring system to determine if any of the produced items contain known defects. In this case, the Causaloid relies on context and external data. Therefore, the Effect Propagation Process emits a flexible knowledge ontology by relying on one or more contexts and potentially multiple external data sources.

6.1.2 Knowledge Derivation

The EPP derives knowledge by applying data to the Causaloid that models that causal relationship to determine whether the causal relation holds true within the applicable context. Consequently, multi-stage reasoning maps directly to the topology of the EPP itself because each effect from a Causaloid propagates further through the EPP topology, which is the structure of the EPP manifested as all connected Causaloids. From this perspective, a "line of reasoning" literally becomes a pathway through the EPP topology. Through the topological approach of knowledge derivation, the Effect Propagation Process provides a flexible way to model complex, contextual, multi-causal domains.

6.1.3 Justification of Derived Knowledge

Discerning the truthfulness of knowledge is one key element of epistemology. The EPP with its explicit context, explicit causal function (Causaloid), and explicit support for external data provides all prerequisites to support the full explainability of each inference. Furthermore, in the case of multi-stage reasoning, the sequence of applied Causaloids establishes the order of explainability. Fundamentally, the EPP leads to explainable causal inference because of complete data, context, and inference function when assuming a static EPP. For a dynamic or emergent EPP, explainability might not be guaranteed for all potential state transitions. An implementation of the EPP has to specify the exact details to support explainable inference and where to establish sensible constraints on explainability.

The gravitas of the EPP epistemology roots in its flexible, contextualized ontology, a powerful knowledge derivation mechanism, and its intrinsic support of explainable causal inference. The epistemology varies depending on whether the EPP process is static, dynamic, or emergent.

6.2 Static EPP Epistemology

For a static Effect Propagation Process, the knowledge is explicitly modeled during the design stage and confined within the context. The quantitative nature of explicitly modeled context and EPP leads to the positivism of the resulting epistemology.

Static context

A static context emits an invariant structure after it is defined; therefore, a static EPP combined with a static context allows for the strongest deterministic guarantees, albeit at the expense of flexibility. Static contexts remain an invaluable tool to model contextual data that remain structurally invariant, which is a common situation when integrating external data sources. The content, structure, richness, and accuracy of that static context profoundly determine the epistemology of what can be known through the EPP.

Dynamic context

In a dynamic context, the context structure itself evolves, e.g., new elements (i.e., a quarter of a year) are added as the data feed progresses. By definition, a dynamic context relies on a generating function to gauge the dynamic changes of the context. The impact on the epistemology of a static EPP remains minimal, though. Fundamentally, dynamic contexts are used when structural elements occur at either regular intervals or otherwise determinable occurrences, and therefore, the EPP can model these elements regardless of whether they have been added to the context yet. For example, a Causaloid that determines whether the sum of the previous three monthly financial reports matches the quarterly financial reports for the current quarter might be a precondition if the "current" quarter in the context has been updated. Therefore, dynamic contexts simplify domain modeling while leaving the epistemology modeled in a static EPP intact.

6.3 Dynamic EPP Epistemology

For a dynamic Effect Propagation Process, the dynamics are captured in generative functions that evolve either the EPP, the context, or both. Conceptually, these generative functions could range from deterministic, rule-based algorithms that construct or modify Causaloids and Context structures based on predefined logic or specific triggers, to more adaptive mechanisms. For instance, a generative function for a dynamic Context might be a higher-order function that, given the current state and new inputs, returns an updated Context graph, a practice well established in functional programming to build dynamic systems.

The ontology of knowledge may evolve as a result of the evolving EPP, and the impact on the epistemology is profound, depending not only on the EPP evolution itself but also on the interaction with its context, as it can happen that both the EPP and its context evolve dynamically.

Static context

For a dynamic EPP, a static context may serve as the foundational layer that captures core data that remain structurally invariant. As with a static EPP, the static context fundamentally determines the epistemology of what can be known through the dynamic EPP.

Dynamic context

For a dynamic context, though, the impact on the epistemology captured in a dynamic EPP can be profound. For example, with the advent of model context protocol (MCP), which lets LLMs call into tools to retrieve or modify data, a causaloid in a dynamic EPP may trigger an MCP invocation, which then updates the context by expanding its structure, and then triggers a generative function that creates a new Causaloid based on the retrieved contextual data, which then analyzes either a newly created part of the context or a new external data feed created by the MCP. As a consequence, the epistemology in this case depends on a dynamic EPP-Context co-evolution.

Dynamic Co-Evolution

When both the EPP and the engulfing context evolve dynamically in what can be seen as a co-evolution, then no fixed epistemology can be established anymore because the inference based on a generated Causaloid over newly added sub-structures of the context may or may not occur depending on the occurrence of the underlying trigger event(s).

More profoundly, it might not be possible any longer to use automated explainability to discern the appropriateness and relevance of the generated Causaloids and contextual shifts in response to external changes. This introduces an element of interpretivism to the resulting epistemology: the derived knowledge requires the observer to apply a conceptual framework for understanding the system's complex and dynamically evolving behavior.

6.4 Emergent EPP Epistemology

Unlike a dynamic EPP, an emergent EPP does not evolve anymore based on pre-determined triggers that initiate pre-defined generative functions. Instead, an EPP is considered emergent when the underlying generation process leads to novel causal configurations, reasoning pathways, or new generative principles for Causaloid-context interactions that were not explicitly encoded beforehand. The generation process may incorporate principles from evolutionary computation, novelty search, or machine learning embedded in the EPP itself. While the full exploration of AI-driven generative functions for the EPP remains future work, the foundational idea of using programmable functions to dynamically define and evolve both the EPP and its context is a natural extension of the EPP's core design philosophy.

Regardless of the mechanism, an emergent EPP does not interact with its Context using pre-defined procedures but instead relies on procedures generated by the EPP itself. The key indicator of emergence is that its novel behavior was not foreseeable by its initial designers.

Static context

Like a static or dynamic EPP, when the static context has been defined upfront, it fundamentally determines the epistemology of what can be known through the dynamic EPP within the contextual boundaries.

Unlike a static or dynamic EPP, an emergent EPP may or may not generate a new static context and that indeed alters the epistemology emerging from an emergent EPP.

Dynamic context

Likewise, when an emerging EPP creates or modifies a dynamic context, the emerging epistemology cannot be determined any longer because of the resulting co-emergence of the EPP and its context.

Dynamic Co-Emergence

An EPP co-emerges with its context when the underlying generation process leads to novel causal configurations that were not explicitly encoded beforehand. This can happen when the EPP contains methods of machine learning that evolve the EPP itself in response to a dynamically changing context. As a result of the dynamic, non-deterministic self-modification of the EPP itself, the spectrum of subsequent factual representations in the context and the emerging causal structures cannot be predicted any longer.

Therefore, determining the truthfulness of the emerging causality imposes a non-trivial challenge that adds an elevated level of pragmatism to the epistemology. The pragmatism becomes necessary because it is not guaranteed that the underlying dynamic context always leads to a truthful representation of the world it seeks to model, but the generated causal relationships may not always be correct either. Both can happen due to generative errors during the EPP. Generative errors may result from complex interactions that contain steps that, in isolation, are correct, but when combined in a certain order may lead to an incorrect outcome. This is a typical characteristic of increasingly complex dynamic systems that need to be considered by taking an operational stance on truth.

6.5 Operationalized Meaning of Truth

The meaning of truth evolves depending on the modality of the EPP because the underlying reference for a true statement varies depending on the chosen modality. Per the definition of knowledge, a true belief must entail a high degree of justification and come from a reliable source to count as knowledge. It is the underlying justification process that depends on the modality of the EPP that causes the shift in the meaning of truth to vary.

For a static Effect Propagation Process, the meaning of truth aligns with the classical correspondence theory. This means that if the context encodes accurate facts and the causal relationships are true, all derived forms of knowledge must be true. Justification rests upon the verifiable mapping between the EPP's explicit model encoded in its context and the part of reality it purports to represent. The static EPP implicitly operates under the assumption that its model is a faithful mirror of objective facts. Here, the truth of an inference is determined by the adherence to the contextual facts and encoded causal relationships. As a result, a static EPP leads to deterministic verifiability within the confined boundaries of its context. As the EPP transitions into a dynamic modality, the meaning of truth begins to shift towards a coherent adaptability to dynamic interaction with a changing context. A dynamically modified causal relationship is deemed true if it maintains consistency with the facts in its evolving context. In a dynamic modality, the justification of knowledge becomes contextually and temporally aware. Therefore, truth is assessed by the EPP's capacity to maintain a relevant and internally consistent causal understanding while navigating a temporal dynamic context. This leans towards a coherence theory of truth, where coherence itself must be evaluated relative to the EPP's intricate temporal structures.

For a contextual co-emergent EPP, the meaning of truth shifts further toward pragmatic efficacy. This shift becomes necessary because of the emergence of relativistic causal relationships from the EPP that co-evolve with its context. Here, establishing an objective a priori truth becomes elusive since the fabric that would traditionally serve as a stable reference for truth is itself emerging dynamically alongside the causal inference made from it. Instead, the truth of an emergent causal inference is established by its utility in enabling the EPP to navigate its environment within its temporally complex context. This pragmatic efficacy means that truth, defined by its functional value, becomes inherently system-relative and context-dependent.

Indeed, a functional value could serve as a fitness function guiding the emergent process itself, thus raising fundamental questions about alignment. Consequently, pragmatic efficacy can lead to multiple, functionally 'true' yet distinct causal understandings, each valid within its own emergent trajectory and its relativistic interrelation with its context.

This dynamic interplay, where the EPP generates both its context and the Causaloids that encode the causal relationships that operate within that context, presents a research opportunity. It allows for the exploration of relativistic emergent causality and how coherent and pragmatically effective causal understandings can arise in systems that lack a fixed predefined spacetime. This might be of interest in theories of fundamental physics where spacetime itself may be an emergent phenomenon arising from more fundamental processes.

6.6 Causal Emergence

The problem of modeling systems with no a priori causal structures motivates a different view of causality that sets the stage for tackling causal emergence. The Effect Propagation Process framework's detachment from fixed spacetime and its focus on a generative function establish the foundation for causal emergence. Static causal discovery, for example, Pearl's DAGs framework, assumes a fixed causal structure and thus cannot handle causal emergence. Granger causality assumes that time-dependent variables change, but the causal structure remains fixed and therefore cannot handle causal emergence either. This argument holds true for any dynamic system with fixed causality because of the inability of the underlying methodology to handle spacetime-agnostic causal structure. The Effect Propagation Process instead proposes that the causal relationships themselves emerge, change, and may even disappear. The implications of this approach lead to a fundamental reassessment of how to operationalize causality:

Causal dynamics

The inspiration from causality in quantum gravity was carefully chosen because of its unique ability to reconcile dynamic and static structures and its handling of deterministic and probabilistic modalities. The underlying idea in quantum gravity is that the spacetime fabric of reality itself emerges from an underlying process. While we do not have the scientific methods and technology to verify this idea on the quantum level, we can carefully, within boundaries, transfer the idea to the EPP notion that causality itself emerges from an underlying generative process and, therefore, model the dynamics of causal emergence. The properties of EPP become apparent when looking through the lens of modeling the dynamics of causal emergence.

- **Temporal order becomes irrelevant:** Because causal relations can emerge from an underlying process.

- **Spacetime-agnostic becomes necessary:** Because the generative process is concerned with establishing relations of effect propagation, the exact fabric through which those effects propagate is conceptualized as an external context; therefore, the EPP itself has to be spacetime-agnostic.

- **Hardy's Causaloids are necessary:** Because the EPP itself is spacetime-agnostic, a different representation of causal relationships that is also spacetime-agnostic becomes necessary and the causaloid proposed by Lucian Hardy has been deemed the best fit.

- **Centrality of the generative function:** Because the EPP can represent causal relations as either static, dynamic, or emergent, the generative function takes on a central role to express those causal relations. Furthermore, a generative function may generate the engulfing context as a specific fabric for the effect propagation process.

6.7 Discussion

The epistemology of the Effect Propagation Process is a direct consequence of its architectural design. By externalizing the environment into a computable Context and defining causality as a composable Causaloid, the EPP creates a spectrum of epistemic possibilities. Each step along this spectrum, from the static to the emergent, represents a deliberate trade-off between verifiability and adaptability, fundamentally altering how knowledge is derived, justified, and what it means for a belief to be "true."

In a Static EPP, the epistemology is one of positivism. Knowledge is a mirror of a pre-defined reality. With a fixed causal model and a static Context, the system's reasoning is deterministic and its conclusions are verifiable against a stable ground truth. This modality provides the strongest guarantees of correctness, making it suitable for well-understood, closed-world problems. As we introduce dynamism, first in the Context and then in the causal structure itself, the epistemic ground begins to shift.

In a Dynamic EPP, where the model co-evolves with its context, the epistemology becomes one of coherence. Truth is no longer a simple correspondence to a fixed, external state, but the internal consistency of the causal model with an ever-changing stream of information. A belief is justified if it remains part of a coherent causal explanation of the world. This introduces an element of interpretivism, as the "truth" of the system's knowledge depends on its continuously evolving conceptual framework.

Finally, in an Emergent EPP, where the rules of evolution are themselves subject to change, the epistemology becomes one of pragmatism. When the system can generate novel causal structures not foreseeable by its designers, the classical notion of an objective, a priori truth becomes untenable. Knowledge is justified by its functional value: a belief is "true" if it leads to a successful outcome in navigating the system's complex environment. This shift to pragmatic efficacy, where truth is system-relative and context-dependent, is the necessary epistemic price for the EPP's ultimate adaptability. However, if we can no longer guarantee that a system's actions are based on what is verifiably true, we must instead ensure that the system's action are aligned with what is verifiably intended. This necessary shift from epistemology to teleology is the subject of the next chapter.

7 The Teleology of the Effect Propagation Process

7.1 Overview

The preceding chapters have established the Effect Propagation Process as a conceptual and formal framework for modeling dynamic causality. Higher-Order Emergence provides a formal language for describing systems capable of recursively evolving their own causal and contextual structures. However, the concept of emergence gains enormous expressiveness to handle dynamically evolving situations, but at the expense of determinism. The loss of determinism leads to a profound challenge of alignment because, when context and causal structure dynamically co-evolve, how do we ensure the resulting system states remain aligned with mission objectives? The metaphysics of the EPP already established that neither verification nor alignment is possible without an additional verification mechanism. In response, the EPP establishes an effect ethos as a mechanism to verify alignment with mission objectives or codified values to ensure that, regardless of dynamic modalities, the system always operates within defined parameters. The EPP and its implementation DeepCausality provide a synergistic foundation for the effect ethos, because:

- Real-world safety problems are not confined to simple geometries. Safety systems for avionics, maritime, and robotics require native support for non-Euclidean geometries.

- Ethics deals with moral dilemmas that evolve dynamically. As a complex situation changes, so do priorities. Therefore, the Effect Ethos needs access to a dynamically evolving context to resolve conflicting objectives by adjusting priorities relative to the context.

- Ethics aims to prevent unwanted consequences. Thus causal state machines are required to have an intermediate step that applies ethics during the translation of reasoning insights into actions to prevent actions that would violate the encoded ethos.

The EPP already provides a solid foundation for the effect ethos by providing context, access to complex geometries, and causal state machines. However, the metaphysics in section 5 of the EPP also established three fundamental crises, of truth, justification, and explainability, as fundamental properties of higher-order emergence. The crisis of truth has proven to be particularly problematic as elaborated in the epistemology in section 6. The metaphysics of axiology establishes two new primitives for a normative framework that shifts the anchor away from an ambiguous epistemology (what is true) to clear and decidable teleology (what is its intent) to mitigate the three identified crises:

- **The Teloid:** A computable unit of purpose. Functioning as a prospective guard of intent, a Teloid is a verifiable function that intercepts a proposed action from a Causal State Machine and evaluates it against a defined goal or policy before execution. This introduces a real, deliberative step of teleological verification against stated intent deeply integrated into the system's core reasoning engine. The Teloid ethical decisions are relative to the system's current context to ensure correct contextual priorities.

- **The Effect Ethos:** A prospective validation framework that ensures alignment before action. The Effect Ethos is the inference engine that operates on a collection of Teloids. Before any Causal Action is executed, the Effect Ethos intercepts the proposes action and uses the DDIC calculus to evaluate it against all relevant norms. This provides a holistic, composable, and mechanistic framework for verifying that a proposed action complies with fundamental principles such as safety, fairness, or regulatory constraints.

Combined, the Teloid and Effect Ethos form an architecture within which ethics becomes a computable. The distinction between a prospective "Teloid" (guarding actions) and a retrospective "Effect Ethos" (validating outcomes) exists for a specific reason. A proposed action is vetted upfront against a set of codified rules to prevent catastrophic failures before they can happen. However, a reasoning outcome, especially when the reasoning is conducted throughout a complex causal hypergraph connected to multiple static and dynamic contexts, can only be evaluated after completion. Many real-world ethical dilemmas involve balancing a locally "correct" action (which a Teloid might permit) against a holistically undesirable emergent outcome that the Effect Ethos may prevent. For instance, a series of individually-approved financial trades could, in aggregate, run against global risk management or potentially violate complex regulatory requirements. In that case, the Effect Ethos prevents the violation before it can happen.

The central challenge faced by the effect ethos is the translation of abstract, relative, and often ambiguous principles such as "safe," "fair," or "efficient" into a verifiable and computable format. Complicating matters further, operational rules often have a hierarchical order where some rules override others, but only in some defined cases. Therefore, contextual conflict resolution is assumed to be the norm and needs to be addressed accordingly.

7.2 Defeasible Deontic Foundation

Addressing the identified challenge, the foundation of the effect ethos is directly inspired by the Defeasible Deontic Inheritance Calculus[7] (DDIC) pioneered by Olson, Salas-Damian, and Forbus. The DDIC is an axiomatic system designed for the purpose of resolving conflicts between evolving norms in a dynamic context and therefore it fulfills the core EPP requirements. The DDIC defines a norm as a formal tuple (Agent, Behavior, Context, Deontic Modal) and its rules are explicit, symbolic inference rules

The DDIC formalizes two conflict resolution heuristics: Lex Posterior (the later rule wins) and Lex Specialis (the more specific rule wins). Critically, it achieves conflict resolution "defeasible inheritance", which means a later or more specific norm can defeat the inheritance of a more general or earlier rule. The DDIC norm tuple explicitly includes a parameter for Context and thus ensures contextual relevance for conflict resolution. Because DDIC is a formal calculus, its reasoning process is traceable. A decision to permit or forbid an action can be explained by providing the chain of axioms and defeaters that were triggered[7].

7.3 Encoding Ethics in a Teloid

The DDIC defines a norm as a formal structure representing an ethical rule. The EPP adapts the formal structure of the DDIC and maps its components directly to the EPP architectural primitives. A Teloid is, therefore, the concrete instantiation of a single DDIC norm. The mapping of the DDIC components is shown in table 4.

Table 4: Mapping DDIC components to the EPP

DDIC	EPP	Description
Agent	Model	The EPP model itself is the agent whose behavior is being governed.
Norm	Teloid	The teloid represents a single DDIC norm
Deontic Modal	Teloid Type	The normative status: Obligatory, Impermissible, or Optional
Behavior	Causal Action	The proposed action from the CSM being evaluated
Context	Context Query	A query against the Context

The calculus of defeasible inheritance enables the EPP to reason efficiently about the implications of its norms. For example, a Teloid forbidding a general behavior will cause its normative status to "inherit" downwards, making more specific behaviors also impermissible. Crucially, this inheritance is "defeasible," meaning it can be blocked or defeated by a more specific or more recent Teloid. For instance, a general permission to change lanes can be defeated by a newer, more specific prohibition against doing so in a construction zone. Furthermore, to handle nuanced decisions between permissible actions, a Teloid with an Optional modal can also contain an associated cost function. This cost function, evaluated against the Context, provides a quantitative measure used by the Effect Ethos to adjudicate between multiple non-forbidden options, ensuring that even optional behaviors are selected in a resource-aware and efficient manner. This provides a formal, predictable, and traceable mechanism for handling exceptions and resolving conflicts.

7.4 Deontic Inference

The Effect Ethos conducts deontic inference via the the DDIC calculus using a multi-step process that is deeply integrated into the foundation of the EPP:

- Interception and Contextual Filtering
- Belief Inference via Inheritance
- Conflict Resolution via Defeasibility
- Verdict Finding via Ethical Consensus

Interception and Contextual Filtering

When a Causal State reaches a conclusion that would activate a Causal Action, the Effect Ethos intercepts the Causal State to determine its teleological scope. This scope defines the nature and extent of the ethical review required. For non-critical actions, the scope may be empty, allowing the action to proceed without review. For critical actions, the Causal State defines its teleological scope through one or more tags. The tagging mechanism is a cornerstone of the EPP's modularity. Teloids are independently authored with their own corresponding tags, allowing them to function as general-purpose, reusable norms. A single Teloid for battery saving, tagged low_power, can be applied to any Causal State that triggers a power-related action, preventing the creation of overly specialized rules.

The Effect Ethos uses the tags from the intercepted Causal State to select a relevant subset of all Teloids for the ethical assessment. It then proceeds to query the Context only for this smaller, relevant set, ensuring it has the latest contextual data for its evaluation and can identify any Teloids that have become stale in case of missing context information. This two-level filtering, first by tag, then by context, ensures that the subsequent deontic inference is highly relevant, computationally efficient, and fast enough for real-time applications.

The causal state designates the applicable teloids to decide the authorization of its conclusion. The Causal State Machine, see section 4.10, separates the decision to activate an action from the action itself to allow for complex decision logic. The Effect Ethos uses the mechanism as intended because the decision whether an action is permissible can only be made upfront. Because the applicable Teloids are designated in the causal state, the Effect Ethos only verifies a smaller but relevant subset of all Teloids thus accelerating the decision process.

Belief Inference via Inheritance

The Effect Ethos then takes this active set of Teloids and applies the DDIC's inheritance to infer a set of normative beliefs. It forward-chain reasons from the active norms to derive all their implications. It is expected that conflicting normative beliefs will occur frequently at this stage because of the dynamic nature of the EPP reasoning and the dynamic context.

Belief Inference via Inheritance provides a formal, predictable, and traceable mechanism for handling exceptions and resolving conflicts. To manage these relationships explicitly, particularly direct defeaters where one rule is a designated exception to another, the Effect Ethos organizes Teloids into a Teloid Graph. This graph structure, formally defined in Chapter 10, makes the inheritance and defeat pathways between norms an explicit and verifiable part of the ethical model.

Conflict Resolution via Defeasibility

As the Effect Ethos infers new beliefs, it simultaneously checks the "defeater" conditions defined in the DDIC's axioms. The DDIC fomalizes two mechanisms for norm conflicts resolution. However, the DDIC acknowledges that Lex Superior, the rule with the highest priority wins, as a key strategy for norm conflict resolution, but does not formalizes it in its calculus most likely because a norm's priority is an external property that, unlike the other two modalities, cannot be derived from the logical content of the norms themselves. The EPP, does include Lex Superior for its practicality to simplify a normative order and give more design flexibility. In total, the EPP supports three modalities for normative conflict resolution:

Lex Posterior (The More Recent Rule Wins): This principle is based on the time a norm was stated. If two norms conflict, the one with the later timestamp is given precedence and defeats the earlier one[7].

Lex Specialis (The More Specific Rule Wins): This principle states that a more specific norm should override a more general one. The DDIC establishes that in many cases, what appears to be a conflict resolved by Lex Specialis is actually just a natural consequence of deontic inheritance, where a more specific rule adds an exception or refinement to a general one without creating a true logical conflict[7].

Lex Superior (The Highest Priority Rule Wins): This principle is based on the priority property of the norm with the and establishes that the rule with the highest priority is given precedence and defeats the rule with the lower priority.

The conflict resolution step uses these three principles to systematically eliminates contradictions and to producing a final, coherent, and conflict-free set of normative beliefs.

Verdict Finding via Consensus

The Effect Ethos examines this final set of norms and seeks a verdict via a consensus. While the DDIC proves that, through defeasible inheritance, it can resolve the logical contradictions to yield a conflict-free set of norms, it does not have a mechanism of reaching consensus among all remaining norms. The EPP addresses this by establishing an order based mechanism to reach a consensus based verdict. The uses a clear precedence ($Impermissible > Obligatory > Optional$) to handle the output of a the previous deontic conflict resolution.

If the proposed Causal Action is present with a single norm that yields impermissible, the Effect Ethos returns a Forbidden verdict, providing the specific Teloid(s) that led to the prohibition as a justification. In practice, it is expected that the set contains a mixture of modalities i.e., some are obligatory and must be adhered to, some might be impermissible, and others might be optional. Thus the consensus rules are as following:

The first consensus rule is that impermissible overrides all other modalities. The second consensus rule is that obligatory overrides optional. And the third consensus rule is that optional supports either of the other ones when its associated cost warrants it. In case the final consensus reaches an optional conclusion, the total associated cost must

fall below a defined threshold to convert the verdict into permissible, otherwise it will be designated as impermissible due to unjustifiable associated cost. In case the final verdict reaches an obligatory consensus, it then provides the specific Teloid(s) as a justification for the permission.

The Effect Ethos makes ethics a first-class component of the EPP that is rooted in the formal DDIC calculus. As a result, the reasons for any decision are fully transparent, traceable and auditable.

7.5 Discussion

The Teloid and Effect Ethos are directly recognizable as "computable policy" and "auditable safety layers" that broadly translate into two new categories:

- **Compliance-as-Code:** The idea of modular Teloids for regulations (e.g., a "Reg-T Teloid") that could be audited directly would lower regulatory risk (fines) and operational cost (standardization).
- **Verifiable Safety for Autonomous Systems:** This provides a concrete architecture for satisfying safety standards (like ISO 26262 for automotive), which is currently a major challenge for any autonomous systems.

One practical application of Compliance-as-Code would be the formal verification of adherence to regulatory requirements directly embedded into the model itself. It is not unthinkable that regulators might want to see audits of the codifying teloids as a means to ascertain and monitor regulatory compliance. Another practical application is the development of modular reference Teloids that codify specific regulations for certain domains with mandatory industry rules, for example in finance, to lower the cost of compliance. For autonomous systems, embedding specific safety rules becomes not only streamlined, but easier to audit, verify, and simulate. Lastly, while neither the Teloid nor the Effect Ethos can directly answer the question of whether a specific inference or proposed action is the right thing with respect to its context, at least these are feasible primitives to build a solution to answer those questions.

Challenges will arise mostly from formalization and verification of the proposed Teloid and Effect Ethos. Specifically, at least the following questions need to be addressed in future development:

- How do we formally verify the Teloid itself
- How do we prove that a composite Effect Ethos is complete and covers all necessary edge cases?
- How do we prove, even if a composite Effect Ethos is correct, that it will be deterministically applied?

The Teloid and Effect Ethos are presented as future work because addressing these immense challenges clearly falls outside the scope of the presented EPP, but still warrant further consideration. While the formalization is a subject for extensive future work, the implementation can re-use existing concepts and primitives already built in DeepCausality and thus substantiate the feasibility of the proposal.

The capability for higher-order emergence carries the risk of uncontrolled or undesirable system evolution. The "Crisis of Truth" is not a theoretical abstraction but a practical safety concern. The proposed architecture of the Teloid and Effect Ethos is the primary mechanism for managing the risks that result from dynamic emergence. The Teloid can be engineered to constrain the generative process by rejecting proposed structural modifications that violate predefined safety, ethical, or operational policies. However, no set of prospective rules can be proven complete. The retrospective Effect Ethos provides a second layer of defense, assessing holistic outcomes where individually correct actions might lead to an undesirable emergent state.

It is crucial, however, to recognize the pragmatic reality of applying EPP: real-world systems will be hybrid models. The majority of their components will be static or governed by predictable dynamics. Only a small but critical subset of the system will be designed to be truly emergent. Traditional brute-force testing is computationally infeasible due to combinatorial explosion. Likewise, formal verification, while powerful for deterministic systems, may not be applicable to a system whose state space can evolve dynamically relative to a dynamic context. The most viable and rigorous path forward is adversarial stress-testing of the teloids and effect ethos. It is possible to systematically search for emergent loopholes and stress-test the Effect Ethos by using Deep Reinforcement Learning to intelligently and adversarially explore the state space of the learned world model.

Adversarial stress-testing does not offer absolute safety guarantees. The potential for unforeseen behavior in a sufficiently complex system remains, as risk is intrinsic to the nature of dynamic emergence. It represents, however, a principled and practical engineering discipline for managing that unavoidable risk. The alternative is to either forgo the benefits of adaptive dynamic systems or to deploy them without a comparably rigorous validation strategy. The proposed Teloid and Effect Ethos, validated through adversarial stress-testing, serve as the tools for navigating causal emergence responsibly.

Managing the intrinsic risk of emergent causality is not a challenge for a single methodology; the problem represents an ongoing challenge for the fields of AI safety, formal verification, and causality. The EPP, with its transparent and auditable architecture, is therefore offered as a high-fidelity testbed for exploring these foundational issues. The author acknowledges that the exploration of causal emergence requires deep inquiry, probing questions, and different perspectives from a multitude of diverse stakeholders. The transparent and open-governance of the DeepCausality project, hosted at the LF AI & Data Foundation, provides a vendor-neutral venue for facilitating such an essential discussion.

8 The Ontology of the Effect Propagation Process

8.1 The Ontology of Being

The static ontology defines the concrete primitives that are the fundamental building blocks, the "matter" (hyle) and "form" (morphe) of the EPP, as defined by the Metaphysics of Being. Each primitive is a direct instantiation of the Monoidic Primitives and Isomorphic Recursive Composition, designed to represent distinct categories of existence within the EPP's universe. This section details these foundational entities, which, when combined, form the foundation upon which all dynamic processes and emergent phenomena are built.

8.1.1 The Causaloid: A Unit of Causality

Causality is fractal. A high-level cause (e.g., "economic recession") can be decomposed into a network of smaller, interacting causes (e.g., "inflation," "supply chain disruption," "interest rate hikes"), each of which can be further decomposed into smaller causes. Classical causality has always struggled with this reality because structural composition is fundamentally at odds with both the SCM representation and the causal DAG.

The EPP proposes to represent causality closer to its fractal nature: The Causaloid. In line with the EPP metaphysics, the Causaloid is a monoid.

The identity element of the monoid is the Singleton 'Causaloid'. This is the fundamental unit of causality. It has an id as identity and a causal function that captures the causal relationship. The form of the Causaloid is isomorphically recursive to enable uniform composition. The causaloid is isomorphic in the sense that a causaloid can have different types and yet share the shape of a causaloid. For example, a causaloid can be:

1. A singleton causaloid that is a single cause.
2. A collection of causaloids.
3. A graph in which each node is another causaloid.

Applying the concept of isomorphically recursive composition means that any number of causaloids of different types uniformly compose. A complex causal graph can be encapsulated into a single Causaloid, which then becomes a node in another causal graph that itself is part of a causal collection. This creates a mechanism where:

- Complexity is manageable
- Scalability is inherent
- Abstraction is first-class

Complexity is manageable because each causaloid can be built and tested in isolation, or as part of a specific subgraph. The mechanism scales from a singular unit to larger collections up to complex graphs. While the system complexity scales, the conceptual overhead remains constant because a single concept, the causaloid, scales from simple to complex, from small to large. Abstraction is first-class because causal reasoning happens uniformly regardless of the underlying complexity. The reasoning mechanism remains the same regardless of whether a causaloid is a singleton, a collection, or a graph.

The monodic binary operation for composition results from the fact that causaloids are isomorphic, which allows to combine a causal collection into a new singleton causaloid. Likewise, an arbitrarily complex causal graph composes uniformly any number of causaloids as its nodes and is in itself a causaloid that composes with other causaloids.

Isomorphic Recursive Composition for causality defines an ontology where the distinction between "part" and "whole" is fluid. Every "whole" (a composite Causaloid) can become a "part" in a larger whole without ever changing its fundamental nature as a Unit of Causality.

8.1.2 The Contextoid: A Unit of Context

The Contextoid represents the monoidic primitive of state, an atomic, non-recursive, and identifiable unit of factual information. The EPP ontology makes a strict distinction between these two primitives: Causaloids are recursive and represent reasoning; Contextoids are isomorphic and represent the ground truth upon which reasoning operates. The ontology of the Contextoid does not enforce a single, fixed representation for concepts like "space" or "time." Instead, it defines abstract categories:

- Datoid - A monoidic unit of data-like context.

- Spaceoid - A monoidic unit of space-like context.
- SpaceTempoid - A monoidic unit of spacetime-like context.
- Symboid - A monoidic unit of symbol-like context.
- Tempoid - A monoidic unit of time-like context.

The categorical isomorphism leads to the most critical design principle of this primitive: Contextoids are structurally non-recursive. A Tempoid cannot contain another Tempoid; a Spaceoid cannot contain another Spaceoid. The structural prohibition of recursion guarantees the logical consistency of the Context. It makes it impossible to construct a paradoxical state, such as a time loop where a moment in time is defined as preceding itself. By ensuring the factual bedrock is non-paradoxical and acyclic, the EPP provides the stable, non-contradictory ground truth required for sound, higher-level reasoning to operate.

Conversely, the same categorical isomorphism enables heterogeneous composition within each category. For example, the Spaceoid category can represent a point in a flat, Euclidean space alongside another representing a point in a non-Euclidean GeoSpace. From the perspective of the ontology, both are valid "spatial facts," i.e., instances of a Spaceoid and thus uniformly treated as space-like. This categorical isomorphism enables the EPP to model heterogeneous systems where different parts of reality demand different mathematical representations of space, time, spacetime, symbol, and data. However, the absence of isomorphic recursion deprives the contextoid of its structure.

8.1.3 The Context: A structure for Contextoids

The Context is the structure that gives meaning through relations to the Contextoids. In the EPP, the Context is a first-class ontological entity: a hypergraph structure that holds all Contextoid primitives and the explicit, typed relationships between them. It is partially monodic in the sense that it has an identity, but unlike other primitives, it has no monoidic composition hence its name "Context" reflects that it is not monoidic and thus does not compose. The deliberate choice roots in the fact that, while its elements, the contextoids, are structurally isomorphic and compose, the engulfing structure, the context, does not compose to prevent the complexity that arises from merging graphs, but more importantly, to prevent incorrect states in this critical structure. Despite this decision, the context, however, is neither singular nor absolute.

Static vs Dynamic Context

A static context is established upfront and is structurally assumed to remain invariant during its lifetime. The utility comes from encoding static knowledge, for example, the ICD-10 medical ontology that is standardized for a given version. A static representation for each version of the ICD ensures there is no mixing of different standards.

A dynamic context is one whose structure is dynamically built and modified during its lifetime. The utility applies to situations where context is fluent but structurally known. For example, a monitoring system that receives data feeds from maintenance drones has to add contextoids for drones coming online and streaming data and, likewise, remove contextoids for drones that get out of range and stop the data stream. The dynamic context can only add, modify, or remove contextoids of known types to ensure strict operational safety.

Single vs Multiple Contexts

The EPP ontology explicitly supports multiple frames of reference via multiple contexts. A causal model can be linked to a primary context representing the current, observable reality, while also having access to any number of additional contexts. These auxiliary contexts can represent simulated worlds, historical states, counterfactual scenarios, or backup data sources for real-time data feeds.

Contextual Counterfactuals

This capacity for multiple contexts establishes the foundation for relativistic contextual counterfactual analysis. It allows the system to ask "what if" questions by creating hypothetical realities (a new extra context), modifying specific Contextoids within them (e.g., "what if temperature were 5

8.1.4   The Evidence: A Unit of Facts

Evidence represents a specific monoidic fact presented to a Causaloid for evaluation that may originate from outside the system, i.e., from a sensor reading, is extracted from a contextoid, or is derived from a previous chain of reasoning from a causaloid. A causaloid uses Evidence as input and the context for supporting data used during the analysis. The output of a Causaloid can then be directly verified against the specific Evidence it received, which is fundamental for explainability in complex systems.

Evidence is a generalized isomorphically recursive container designed to support unified reasoning across multiple modalities:

- Deterministic: Boolean values ("true/false").
- Numerical values: Numbers (e.g., sensor readings).
- Probabilistic values (e.g., confidence scores, likelihoods).
- Contextual Links (a Contextoid within a Context), enabling the Causaloid to access complex, structured, and non-numerical facts.

Evidence is recursive and may contain:

- Maps of other Evidence primitives
- Graphs of Evidence primitives enable a Causaloid to access complex, relational data.

The multi-modal Evidence enables the EPP to reason over the full spectrum of information found in complex systems.

8.1.5   The Propagating Effect: A Unit of Influence

The output of a causal evaluation is the PropagatingEffect, a monoidic primitive of influence. The PropagatingEffect is the operational heart of the "Effect Propagation Process" itself, a unit of influence that travels through the causal graph, from one Causaloid to the next, driving the continuous effect propagation process. The PropagatingEffect is a unified inference outcome across different modalities. It is isomorphic in that it can represent:

- Deterministic effect: A definitive boolean outcome ("true/false").
- Probabilistic effect: A quantitative outcome, such as a probability score or an estimate.
- Contextual Link: A reference to a specific Contextoid within a Context.

For the Contextual Link, a causaloid writes its reasoning outcome into a contextoid and then propagates its "effect" as a Contextual link to direct the next Causaloid to use the structured information in the contextoid for further analysis. This Contextoid then becomes the Evidence for the subsequent step in the reasoning chain, enabling dynamic, data-driven causal pathways for non-numerical representation as required, for example, for causal symbolic reasoning enabled by the symboid contextoid type.

8.2   The Ontology of Becoming: Dynamics

The ontology of dynamics in the EPP is governed by the metaphysical principle of contextual relativity. Contextual means that the significance of a monoidic primitive is derived from its relation to its context. Contextual relativity means that state is an emergent property arising relative to its context with the implication that the same object may have different states when used in different contexts, but, more profoundly, the same object may alter its state when

context itself is relativistically altered and propagates its contextual adjustment to all connected monoidic primitives. Contextual Relativity is expressed in the EPP in two ways:

First, a Causaloid's reasoning is relative to the 'Context' it is evaluated against. The EPP ontology explicitly supports multiple frames of reference for each causaloid. A Causaloid asking "Is the pressure critical?" can return true when evaluated against a Context representing a high-altitude environment and false when evaluated against one representing sea level. The causal logic is the same, but the truth it produces is relative to the frame of reference. This is very much in line with the observer principle in the general theory of relativity.

Second, the state of a Contextoid itself is subject to relativistic forces imbued upon its engulfing fabric. For example, a spacetime contextoid may need to adjust its temporal value for gravity-induced time dilation. A quaternion contextoid may need to adjust its rotation value relative to incoming sensor data. Therefore, the EPP ontology defines two critical operations to handle contextual relativity via two operations:

1. Adjust
2. Update

Adjust means an existing value is adjusted for relativistic effects or corrected for sensor drift. The core property of an adjustment operation is to take a correction value, say an offset, and apply it to the existing value.

Update means an existing value is replaced with a new value, for example, a new sensor reading. The core property of an update operation is to take a data value and replace the current one with the new one.

These two operations, adjust and update, determine the difference between a strictly static contextoid and a dynamic contextoid. A static one is assumed to be invariant throughout its lifetime, for example by holding a set of immutable facts, i.e., a thermal threshold. A dynamic contextoid is one that gets either adjusted or updated, for example when new sensor reading becomes available. A simple causaloid then reads the thermal threshold from one contextoid, and reads the current thermal sensor data from a dynamic contextoid, and determines if it is getting closer to the thermal limit. Critically, when transferring the model to a different operational environment that requires a different thermal threshold, the replacement of one contextoid is sufficient to ensure the correct functionality of the modeled thermal system.

8.3 The Ontology of Emergence

The ontology of Emergence is captured in the mechanism of the Generative Process, a four-stage command-execution cycle that transforms external stimuli or internal states into structural modifications of the EPP itself.

1. The Generative Trigger
2. The Generative Command
3. The Generative Process
4. The Generative Outcome

This four-stage process applies to both first-order emergence and higher-order emergence because of its recursive design. Ontologically, each step results in a specific and verifiable object, which lays the foundation for a principled implementation at a later stage.

8.3.1 The Generative Trigger

Emergence begins with a Generative Trigger. This primitive represents the initial impetus for change, signaling to the EPP that a modification of its internal structure becomes necessary. Triggers can originate from external stimuli, the passage of time, or an explicit external command. A generative trigger can arise from internal states, such as the detection of an anomaly, a deviation from a desired goal, or from a change in external state such as the detection of a fundamental change in the environment, i.e., transiting from day to night. The Generative Trigger acts as the catalyst to initiate the generative process.

8.3.2 The Generative Command

Upon receiving a Generative Trigger, a Generative Command is constructed. This primitive is a declarative, verifiable blueprint of a desired action or structural modification. It is the formal expression of the EPP's intent to alter its own substance. Generative Commands are explicit instructions for change, such as the creation, update, or deletion of a Causaloid or Contextoid, the modification of contextual relationships by adding or removing edges in the hypergraph.

The Generative Command is both isomorphic and recursive. It is isomorphic in that it unifies a diverse set of possible actions under a single primitive type. These actions include, but are not limited to:

- No operation. *NoOp*:
- Commands for managing causal logic: *CreateCausaloid*, *UpdateCausaloid*, *DeleteCausaloid*
- Commands for managing contexts: *CreateBaseContext*, *CreateExtraContext*, *UpdateContext*, *DeleteContext*
- Commands for managing facts within a context: *AddContextoidToContext*, *UpdateContextoidInContext*, *DeleteContextoidFromContext*
- A user-defined command for higher-order emergence: *Evolve*.

The Generative Command is recursive through its Composite variant. This variant allows a single Generative Command to contain an ordered sequence (a "stack") of other Generative Commands. This recursive capability enables the EPP to express complex, multi-step transformations as a single, atomic unit of intent. For example, a single Composite command could specify: "add root node, then add Causaloid A, then add an edge between root and A." This ensures that even intricate sequences of operations are treated as a coherent, atomic transaction, maintaining the integrity and verifiability of the system's emergence.

The Evolve command is one notable outlier in that it is not an atomic command. Instead, Evolve is a meta-command; it is an instruction to replace the mechanism that generates commands itself. By definition, Evolve is no longer guaranteed to be deterministic. If the choice of what to evolve into is not itself deterministically derived, then the system's future behavior becomes non-deterministic and potentially unpredictable. Even if the new command is deterministic, the decision to evolve and how to evolve represents a point where the system's fundamental archê kai aitia of change is altered and it is conceptually unclear what this may entail in practice.

8.3.3   The Generative Process

The Generative Process is the operation responsible for actuating the Generative Command. It interprets the command and translates it into concrete, structural modifications within the EPP's ontology.

This stage represents the materialization of the intended change, where new Causaloids are instantiated, Context graphs are reconfigured, or Contextoids are added, modified, or removed. The Generative Process transforms the abstract intent into tangible alterations of the EPP's substance.

The Generative Process is designed to be robust and auditable. It processes each Generative Command deterministically, ensuring that the system's state transitions are predictable given a specific command. Its primary function is to ensure the integrity of the EPP's internal structure during the process of self-modification.

8.3.4   The Generative Outcome

The generative outcome is a primitive that holds both the modified or generated artifact and relevant metadata about the process itself. It attests to the "what happened" during the process in the form of a log detailing every state transition and asserts whether the outcome is in accordance with the stated intention of the generative command. This explicit record of the system's self-modification means that the EPP maintains a complete and verifiable history of its own evolution; a crucial step for auditability, debugging, and the continuous learning process, providing the ground truth for subsequent development of emergence in complex dynamic systems.

8.4   The Ontology of Action

8.5   Discussion

The ontology of the Effect Propagation Process addresses the limitations of classical causal models in dynamic, complex systems. The EPP's ontology deeply integrates a set of primitives: the Causaloid as a recursive unit of logic, the Contextoid as a non-recursive unit of fact, and the Context as a dynamic, multi-frame relational environment. These primitives, combined with the multi-modal Evidence and PropagatingEffect, form a robust foundation for expressing causality not as a static, linear chain, but as a continuous process of effect propagation.

The EPP's explicit ontological primitives and its four-stage Generative Process provide a unique computational platform for exploring the nature of emergence itself. While the principles of Dynamic State Mutation and Dynamic Structural Evolution offer clear pathways for predictable adaptation, the concept of Dynamic Co-Emergence, through the Evolve command, opens a frontier for investigating self-referential self-modification of dynamic systems.

The Evolve command, as defined within the EPP's ontology, represents the system's capacity to fundamentally alter its own generative principles. This capability, while theoretically profound, introduces complex questions regarding predictability, control, and safety. The EPP's transparent, auditable architecture where every Generative Command and Execution Result is explicit offers an unprecedented opportunity for empirical investigation into these phenomena.

By designing and observing systems that engage in self-referential self-modification, practitioners can gain firsthand insights into the dynamics of emergent behavior, the challenges of maintaining verifiability in adaptive systems, and the potential pathways towards engineering truly autonomous and self-adapting dynamic systems. This area represents a rich vein for future research, where the EPP's foundational principles can be tested and expanded through direct computational experimentation.

9 The Formalization of the Effect Propagation Process

9.1 Overview

The Effect Propagation Process (EPP), introduced as a novel philosophical framework implemented in the Deep-Causality project, fundamentally redefines causality to address the limitations inherent in classical, spacetime-bound conceptions. These limitations become particularly acute when considering insights from contemporary quantum gravity—which suggest emergent spacetime and indefinite causal order—and when tackling the practical complexities of advanced system engineering, such as non-Euclidean data structures, non-linear temporal dynamics, and emergent causal regimes. This work presents a first-principles mathematical formalization of the EPP, thereby establishing a rigorous and unambiguous foundation crucial for its continued theoretical development, consistent computational instantiation, and broader interdisciplinary engagement.

The core of this formalization lies in precise set-theoretic definitions of EPP's key components. We define the Contextual Fabric ($\mathcal{C}$), an explicit and potentially dynamic structure composed of Contextoids and Context Hypergraphs, which can represent diverse Euclidean, non-Euclidean, temporal, and abstract relational information, thus operationalizing EPP's spacetime agnosticism. Subsequently, we formalize Causal Units ($\mathcal{G}$), built from operational Causaloids (χ)—unified entities of testable effect transfer that transcend the classical cause-effect dichotomy—and their composition into recursively isomorphic CausaloidGraphs (G) for modeling intricate causal webs. The Causal State Machine (CSM) is then formalized as the mechanism linking causal inferences to deterministic actions.

Beyond these structural entities, the formalism meticulously details the core dynamics of EPP. This includes the Effect Propagation Process (Π_{EPP}) itself, which describes how the activation states of Causaloids evolve through a CausaloidGraph based on their intrinsic functions, contextual inputs, and defined relational logic. Furthermore, we formalize the Operational Generative Function (Φ_{gen}), which provides the mechanisms for the dynamic evolution and emergence of both the Contextual Fabric and the CausaloidGraph structures, a key feature enabling the modeling of emergent causality and regime shifts. The work also demonstrates how foundational EPP philosophical principles, such as the emergence of temporal order and congruence with classical causality under specific conditions, are naturally reflected within this mathematical framework.

The generalization of classical causality is demonstrated in section 9.11 which provides subsections of how the EPP expresses Pearl'SCM, DBN, Granger, RCM, and CATE methodologies using only EPP native primitives. One key insight is that the explicit context of the EPP simplifies counterfactual reasoning significantly.

By providing this clear mathematical language, the formalization of the EPP aims to enhance the overall clarity, consistency, and testability of the EPP. It seeks to empower researchers and engineers across diverse fields to apply EPP for modeling complex, context-dependent, and dynamically evolving causal phenomena, fostering a more robust and nuanced understanding of causality in both theoretical and practical domains.

9.2 Axioms

The theoretical foundation of the Effect Propagation Process is derived from a single axiom, which is itself a necessary consequence of detaching causality from a presupposed spacetime.

Axiom 1 (of Causality). *Every causal phenomenon is an instance of a spacetime-agnostic functional dependency. Let $\mathcal{C}$ be the universal set of all causal phenomena. The axiom states:*

$$\forall c \in \mathcal{C}, \exists f \text{ such that } c \equiv (E_2 = f(E_1)) \tag{1}$$

where E_1 and E_2 are effects and f is a function. The nature of E and f are deliberately unconstrained.

9.3 Formal Derivation of the EPP Primitives

The core architectural primitives of the Effect Propagation Process are not a collection of independent design choices, but are necessary logical consequences that follow from the single causal axiom. This section provides a formal summary of that derivation.

1. **The Contextual Fabric ($\mathcal{H}$): The Background Condition.**
   A function $f(E_1)$ may be conditioned on a set of background variables, K. The general form of the axiom is thus $E_2 = f(E_1, K)$. A mechanism to hold and provide this background information K is therefore necessary.

   $\implies$ Let the **Contextual Fabric** ($mathcalH$) be this mechanism.

2. **The Causaloid (χ): The Functional Primitive.**
   Since every causal phenomenon c is defined by a function f, a computational primitive to encapsulate f is a logical necessity.

   $\implies$ Let the **Causaloid** (chi) be this primitive, whose primary component is the **causal function** (f_χ).

3. **The PropagatingEffect (Π_E): The Isomorphic Message.**
   A function requires inputs and produces an output. Given the compositional nature of functions (e.g., $g(f(E))$), where the output of one serves as the input to the next, a single, uniform message type is logically necessary.

   $\implies$ Let the **PropagatingEffect** (Pi_E) be this isomorphic primitive serving as input to and output from a Causaloid.

4. **The Causaloid Graph ($\mathcal{L}$): The Structural Composition.**
   A system may contain multiple causal phenomena ($c_1, c_2, \ldots, c_n$) that are themselves causally related. A structure is required to represent the set of Causaloids
   chi_1,
   $chi_2, \ldots,$
   chi_n and the dependency pathways of their 'PropagatingEffect's.

   $\implies$ Let the **Causaloid Graph** ($\mathcal{L}$) be a hypergraphwhere nodes are Causaloids and hyperedges are propagation effects.

The subsequent sections provide the detailed set-theoretic definitions of these primitives.

9.4 The PropagatingEffect Primitive (Π_E)

The **PropagatingEffect** (Π_E) is the isomorphic message that serves as both input and output for Causaloids. It is a polymorphic type representing the disjoint union of several distinct kinds of effects.

Definition 9.1 (PropagatingEffect). *A PropagatingEffect $\pi \in \Pi_E$ is an element of the set defined by the disjoint union:*

$$\Pi_E = \Pi_{None} \cup \Pi_{Det} \cup \Pi_{Num} \cup \Pi_{Prob} \cup \Pi_{Link} \cup \Pi_{Map} \cup \Pi_{Graph} \cup \Pi_{Relay}$$

where:

- Π_{None} *is a singleton set representing the absence of an effect.*

- $\Pi_{Det} = \{Deterministic(b) \mid b \in \{true, false\}\}$ *represents a boolean effect.*

- $\Pi_{Num} = \{Numerical(n) \mid n \in \mathbb{R}\}$ *represents a scalar numerical effect.*

- $\Pi_{Prob} = \{Probabilistic(p) \mid p \in [0, 1] \subset \mathbb{R}\}$ *represents a probability.*

- $\Pi_{Link} = \{ContextualLink(id_C, id_v) \mid id_C \in \mathbb{I}, id_v \in \mathbb{I}\}$ *is a reference to a Contextoid within a Context.*

- $\Pi_{Map} = \{Map(m) \mid m : \mathbb{I} \to \Pi_E\}$ *is a map from identifiers to other PropagatingEffects.*

- $\Pi_{Graph} = \{Graph(g) \mid g \text{ is a graph of } \Pi_E\}$ *represents a graph of effects.*

- $\Pi_{Relay} = \{RelayTo(i, \pi) \mid i \in \mathbb{N}, \pi \in \Pi_E\}$ *is a control-flow directive for adaptive reasoning, instructing the engine to jump to the Causaloid at index i with a new input effect π.*

9.5 Contextual Fabric ($\mathcal{C}$)

The Effect Propagation Process (EPP), being spacetime-agnostic, requires a formally defined structure through which effects propagate and within which causal relationships are conditioned. This structure is termed the **Contextual Fabric**, denoted abstractly as $\mathcal{C}$. The Contextual Fabric is not a monolithic entity but can be composed of multiple, distinct contextual realms, each potentially representing different aspects of a system's environment or internal state (e.g., physical space, temporal scales, relational networks, data streams). This section provides a set-theoretic formalization of this fabric, starting from its most granular component, the Contextoid, and building up to collections of Context Hypergraphs.

9.5.1 The Contextoid (v)

The atomic unit of contextual information within the EPP framework is the **Contextoid**. A Contextoid encapsulates a single, identifiable piece of data, which can be temporal, spatial, spatiotemporal, symbolic, or a general data value.

Definition 9.2 (Contextoid). *A Contextoid v is an element of a set of all possible contextoids $V_{\mathcal{C}}$. It is formally defined as a tuple:*

$$v = (id_v, payload_v, adj_v)$$

where $id_v \in \mathbb{I}$ is a unique identifier, $payload_v$ is the content-bearing element, and adj_v is an optional protocol for dynamic modification.

Contextoid Payload

Definition 9.3 (Contextoid Payload). *The $payload_v$ represents the actual contextual information. It is an element from the disjoint union of several universal sets, each representing a fundamental category of contextual information:*

$$payload_v \in \mathcal{D} \cup \mathcal{T} \cup \mathcal{S} \cup \mathcal{ST} \cup \Sigma$$

where $\mathcal{D}$ is the set of data payloads, $\mathcal{T}$ is the set of temporal payloads, $\mathcal{S}$ is the set of spatial payloads, $\mathcal{ST}$ is the set of spacetime payloads, and Σ is the set of symbolic payloads.

Data Payload

Definition 9.4 (Data Payloads ($\mathcal{D}$)). *Let $\mathcal{D}$ be the universal set of all data payloads. Each element $d \in \mathcal{D}$ is a tuple $(id_d, data_d)$, where $id_d \in \mathbb{I}$ and $data_d$ is a value from a generic domain $\mathcal{V}_D$.*

Temporal Payloads

Definition 9.5 (Temporal Payloads ($\mathcal{T}$)). *Let $\mathcal{T}$ be the universal set of all temporal payloads. Each element $t \in \mathcal{T}$ is a tuple $(id_t, scale_t, unit_t)$.*

- *$id_t \in \mathbb{I}$ is its unique identifier.*

- *$scale_t \in \mathcal{T}_{scale}$, where $\mathcal{T}_{scale}$ is a predefined enumerated set of temporal granularities (e.g., {Seconds, Ticks, SymbolicEvent}).*

- *$unit_t \in \mathcal{V}_T$, where $\mathcal{V}_T$ is a suitable ordered value set (e.g., $\mathbb{R}$ or $\mathbb{N}$).*

The set $\mathcal{T}$ may be partitioned into subsets corresponding to different temporal semantics, such as discrete, continuous, or symbolic time.

Spatial Payloads

Definition 9.6 (Spatial Payloads ($\mathcal{S}$)). *Let $\mathcal{S}$ be the universal set of all spatial payloads. Each element $s \in \mathcal{S}$ is defined by a tuple $(id_s, coords_s)$.*

- *$id_s \in \mathbb{I}$ is its unique identifier.*

- *$coords_s$ is a vector $(c_1, c_2, \ldots, c_n)$ where each component $c_i \in \mathcal{V}_S$ for some value set $\mathcal{V}_S$.*

- *The dimensionality n is given by a function $f_{dim} : \mathcal{S} \to \mathbb{N}$.*

For any subset of $\mathcal{S}$, a metric $d : (\mathcal{S}, \mathcal{S}) \to \mathbb{R}^+$ may be defined, allowing for distance calculations. This structure accommodates both Euclidean and non-Euclidean geometries (e.g., geodetic, rotational).

Spacetime Payloads

Definition 9.7 (Spacetime Payloads ($\mathcal{ST}$)). *Let $\mathcal{ST}$ be the universal set of all spacetime payloads. Each element $st \in \mathcal{ST}$ is a composite structure, formally a pair (s, t), where $s \in \mathcal{S}$ and $t \in \mathcal{T}$.*

Symbolic Payloads

Definition 9.8 (Symbolic Payloads (Σ)). *Let Σ be the universal set of all symbolic payloads. Each element $sym \in \Sigma$ is a tuple $(id_{sym}, repr_{sym})$, where $id_{sym} \in \mathbb{I}$ and $repr_{sym}$ is an element from a domain of symbolic representations $\mathcal{R}_{sym}$.*

This polymorphic design allows the EPP to model and reason about highly heterogeneous and complex contextual information in a consistent and formally grounded manner, closing the gap between abstract theory and practical implementation.

Contextoid Adjustability Protocol

The component adj_v represents optional functions implementing the EPP's Adjustable protocol, allowing for the dynamic modification of a Contextoid's payload payload_v. Let $\mathcal{V}_{\text{payload}}$ be the set of all possible payload values and $\mathcal{V}_{\text{adj\_factor}}$ be the set of all possible adjustment factors. The functions are:

- $\text{update}_v : \mathcal{V}_{\text{payload}} \to \text{void}$: This function replaces the current payload_v of the Contextoid with a new payload value.
- $\text{adjust}_v : \mathcal{V}_{\text{adj\_factor}} \to \text{void}$: This function modifies the current payload_v based on an adjustment factor.

The 'void' return type indicates that these functions perform a stateful update on the Contextoid v. Their absence implies the Contextoid's payload is immutable post-instantiation.

9.5.2 The Context Hypergraph (C)

Individual Contextoids are organized into a structured collection called a **Context Hypergraph**. This hypergraph structure allows for the representation of complex, N-ary relationships between different pieces of contextual information, which is crucial for modeling the intricate dependencies found in real-world systems.

Definition 9.9 (Context Hypergraph). *An individual Context Hypergraph C is defined as a tuple:*

$$C = (V_C, E_C, ID_C, Name_C)$$

*where V_C is its finite set of **Contextoid** nodes (each conforming to Definition 3.1, thus $V_C = \{v_1, v_2, \ldots, v_n\}$), E_C is its finite set of **Hyperedges**, $ID_C \in \mathbb{N}$ is a unique identifier for C, and $Name_C$ is a descriptive name.*

Set of Context Hyperedges (E_C). A Hyperedge $e \in E_C$ represents a relationship among a subset of Contextoids in V_C. It is defined as a tuple:

$$e = (V_e, \text{kind}_e, \text{label}_e)$$

where:

- $V_e \subseteq V_C$ is a non-empty subset of Contextoids in C connected by this hyperedge, with $|V_e| \geq 1$.
- $\text{kind}_e \in \mathcal{K}_{\text{relation}}$ specifies the type or nature of the relationship, drawn from a predefined set of possible relation kinds $\mathcal{K}_{\text{relation}}$ (e.g., $\mathcal{K}_{\text{relation}} = \{$containment, proximity, temporal_sequence, logical_and, synonymy$\}$).
- label_e is an optional descriptive label for the hyperedge (e.g., a string).

The use of hyperedges allows a single relational link (e) to connect an arbitrary number of Contextoids (V_e), enabling the representation of multi-way contextual dependencies.

9.5.3 Context Accessor (ContextAccessor($\mathcal{C}_{refs}$))

To enable Causaloids (defined in Section 9.6) to interact with the Contextual Fabric, a mechanism for querying and retrieving contextual information is required. This is conceptualized as a **Context Accessor**.

Definition 9.10 (Context Accessor). *A Context Accessor, denoted ContextAccessor($\mathcal{C}_{refs}$), is a functional interface that provides read-access to a specified subset of Context Hypergraphs $\mathcal{C}_{refs} \subseteq \mathcal{C}_{sys}$.*

Its operations, denoted abstractly, would typically include functions f_{access} such that:

- getContextoid$(id_v, ID_C) \to v \cup \{\text{null}\}$: Retrieves a Contextoid v by its identifier id_v from a specific Context Hypergraph C (identified by ID_C) within $\mathcal{C}_{refs}$.
- getHyperedges$(v_i, \text{kind}_e) \to \{e_1, \ldots, e_m\}$: Retrieves all hyperedges of a specific kind kind_e that involve a given Contextoid v_i.

The precise set of functions within the Context Accessor depends on the requirements of the Causaloid functions that utilize it. This formalism defines the accessor as a means to obtain relevant contextual data for causal evaluation based on identifiers and relational queries.

9.6 Causal Units and Structures ($\mathcal{G}$)

Having formalized the Contextual Fabric ($\mathcal{C}$) within which effects propagate, we now turn to the formalization of the causal entities themselves. The Effect Propagation Process (EPP) posits that causal influence is mediated by discrete, operational units which can be composed into complex structures. This section defines these Causal Units and Structures, denoted abstractly as $\mathcal{G}$. The central entity is the Causaloid (χ), representing an individual causal mechanism. These primitives are then composed into Causaloid Collections and CausaloidGraphs (G) to model intricate webs of causal relationships.

9.6.1 The Causaloid Primitive (χ)

The fundamental unit of causal interaction within EPP is the **Causaloid**. It is a recursively isomorphic primitive that encapsulates a single, testable causal mechanism.

Definition 9.11 (Causaloid). *A Causaloid χ is a tuple:*

$$\chi = (id_\chi, type_\chi, f_\chi, \mathcal{C}_{refs}, desc_\chi, \mathcal{A}_{linked}, I_{linked})$$

where:

- $id_\chi \in \mathbb{I}$ *is a unique identifier.*

- $type_\chi \in \{Singleton, Collection, Graph\}$ *specifies its structural nature.*

- f_χ *is the **causal function** associated with the Causaloid. Its precise signature depends on* $type_\chi$.

- $\mathcal{C}_{refs} \subseteq \mathcal{C}_{sys}$ *is a set of references to Context Hypergraphs for the evaluation of* f_χ.

- $desc_\chi, \mathcal{A}_{linked}, I_{linked}$ *are optional metadata for description and empirical grounding.*

Causal Function for Singletons ($f_{\chi,S}$) For a Causaloid of type Singleton, the causal function is a direct evaluation mapping an input effect and a context to an output effect:

$$f_{\chi,S} : (\Pi_E \times \text{ContextAccessor}(\mathcal{C}_{refs})) \to \Pi_E$$

9.6.2 The Causaloid Collection ($\mathcal{X}_{coll}$)

A **Causaloid Collection** models causal phenomena that arise from the aggregation of multiple individual causes. It encapsulates a set of Causaloids and an explicit logic for combining their effects. This structure is fundamental for representing many-to-one causal relationships in a concise and manageable way.

Definition 9.12 (Causaloid Collection). *A Causaloid Collection, $\mathcal{X}_{coll}$, is a set of Causaloids:*

$$\mathcal{X}_{coll} = \{\chi_1, \chi_2, \ldots, \chi_p\}$$

Aggregate Logic ($\mathcal{A}_L$) The evaluation of a collection is governed by a specific aggregation rule, which determines how the effects of the individual member Causaloids are combined into a single, coherent outcome.

Definition 9.13 (Aggregate Logic). *Let $\mathcal{A}_L$ be the enumerated set of aggregate logics:*

$$\mathcal{A}_L = \{All, Any, None, Some(k)\} \text{ where } k \in \mathbb{N}$$

Each logic corresponds to a standard logical operator applied to the set of evaluated effects from the collection's members:

- ***All**: Logical conjunction ($\wedge$). The collection is active only if all member Causaloids are active.*

- ***Any**: Logical disjunction ($\vee$). The collection is active if at least one member Causaloid is active.*

- ***None**: Negated disjunction ($\neg\vee$). The collection is active only if no member Causaloids are active.*

- ***Some(k)**: k-out-of-n threshold logic. The collection is active if at least k member Causaloids are active.*

Causal Function for Collections ($f_{\chi,C}$) For a Causaloid of type Collection, the causal function evaluates each member of its collection and aggregates the results using a specified logic.

Definition 9.14 (Causal Function for Collections). *The function is a mapping:*

$$f_{\chi,C} : (\mathcal{P}(\chi) \times \Pi_E \times \mathcal{A}_L) \to \Pi_E$$

This function, $f_{\chi,C}$, takes a set of causaloids (the collection $\mathcal{X}_{coll}$), an input effect $\pi_{in} \in \Pi_E$, and an aggregate logic $l \in \mathcal{A}_L$. It computes a set of intermediate effects $\{\pi_1, \pi_2, \ldots, \pi_p\}$ by applying π_{in} to each $\chi_i \in \mathcal{X}_{coll}$, and then applies the logic l to this set to produce a single, final output effect $\pi_{out} \in \Pi_E$.

9.6.3 The CausaloidGraph (G)

Individual Causaloids are composed into **CausaloidGraphs** to represent complex webs of interconnected causal relationships. A CausaloidGraph is itself a hypergraph.

Definition 9.15 (CausaloidGraph). *A CausaloidGraph G is defined as a tuple:*

$$G = (V_G, E_G, ID_G, Name_G)$$

where:

- V_G *is a finite set of causal nodes.*

- E_G *is a finite set of causal hyperedges, representing functional relationships or propagation pathways between nodes in V_G.*

- $ID_G \in \mathbb{I}$ *(or $\mathbb{N}$) is a unique identifier for the CausaloidGraph.*

- $Name_G$ *is a descriptive name for the CausaloidGraph (e.g., a string).*

Definition 9.16 (Set of Causal Nodes (V_G)). *Each causal node $v_g \in V_G$ is defined as a tuple $v_g = (id_g, payload_g)$, where $id_g \in \mathbb{I}$ is its unique identifier within G. The $payload_g$ embodies the principle of **recursive isomorphism** central to EPP, allowing for hierarchical model construction. It can be one of:*

- *A single Causaloid χ (as per Definition 4.1).*

- *A collection of Causaloids $\{\chi_1, \chi_2, \ldots, \chi_m\}$, where the collection itself might have aggregate evaluation logic (e.g., "active if any χ_i is active").*

- *Another entire CausaloidGraph G', enabling the nesting of causal sub-models.*

Definition 9.17 (Set of Causal Hyperedges). *Each causal hyperedge $e_g \in E_G$ represents a directed functional relationship or pathway for effect propagation. It is defined as a tuple:*

$$e_g = (V_{source}, V_{target}, logic_e)$$

where:

- $V_{source} \subseteq V_G$ *is a non-empty set of source causal nodes.*

- $V_{target} \subseteq V_G$ *is a non-empty set of target causal nodes.*

- $logic_e$ *defines the functional relationship or condition under which effects propagate from V_{source} to V_{target}. This logic might range from simple conjunction/disjunction of source node states to more complex functions that determine how the activation of source nodes influences target nodes.*

This hypergraph structure allows for many-to-many causal relationships between cualoids.

Definition 9.18 (State of the CausaloidGraph (S_G)). *The state of a CausaloidGraph G at any given point in its evaluation is defined by the activation states of its constituent causal nodes.*

Definition 4.3 (State of CausaloidGraph): *The state S_G is a function mapping each causal node in V_G to an activation status:*

$$S_G : V_G \to \{active, inactive\}$$

This state evolves as effects propagate through the graph according to the Causaloid functions and the logic of the hyperedges.

9.7 Effect Propagation, Dynamics, and Emergence

The preceding sections formalized the static structural components of the Contextual Fabric ($\mathcal{C}$) and Causal Units/Structures ($\mathcal{G}$). This section now delves into the heart of the Effect Propagation Process (EPP): its core dynamics, the mechanisms for generating and evolving these structures, and how these formalisms embody the foundational philosophical principles of EPP. This operationalization is crucial for understanding how EPP moves beyond classical causality to address complex, dynamic, and emergent systems.

The Effect Propagation Process (Pi_EPP)

The core dynamic of EPP is the propagation of effects through a CausaloidGraph G, influenced by the Contextual Fabric $\mathcal{C}_{sys}$ and triggered by observations or internal states. This process, denoted Π_{EPP}, describes how the activation state S_G of the CausaloidGraph evolves from an initial state S_G to an updated state S'_G.

Definition 9.19 (Defining an Effect (epsilon) within EPP). *Within the EPP formalism, an "Effect" (ε) is primarily represented by:*

1. *The activation state (active or inactive) of a Causaloid (χ) or a causal node (v_g) within a CausaloidGraph, as determined by its causal function f_χ.*

2. *The transfer of this activation status, or information derived from it, to other connected Causaloids according to the graph structure and hyperedge logic.*

An active state $S_G(v_g) = active$ signifies that the causal mechanism encapsulated by v_g has met its conditions for effect transfer.

The nature of the effect is qualitatively described by $desc_\chi$, and its consequence is how its activation influences other parts of the CausaloidGraph.

Definition 9.20 (Input Observations and Triggers (O_input)). *The propagation process Π_{EPP} is typically initiated or influenced by external inputs or observations.*

Definition 5.1 (Input Trigger Set): Let $\mathcal{O}_{input}$ be the set of all possible input observations relevant to the CausaloidGraph G. An input $o_{in} \in \mathcal{O}_{input}$ is an element compatible with the $\mathcal{O}_{type}$ expected by one or more Causaloid functions f_χ within G. At any given evaluation cycle, a set of specific input observations $O_{trig} \subseteq \mathcal{O}_{input}$ serves as triggers. These triggers might be:

- *Directed at specific Causaloids in G (e.g., root nodes, or nodes representing sensors).*

- *Representing updates to the Context Collection $\mathcal{C}_{sys}$ which, in turn, affect the evaluation of context-aware Causaloids.*

Definition 9.21 (The Propagation Transition Function)). *The propagation of effects within a CausaloidGraph G is formalized as a state transition function Π_{EPP}. This function describes how the graph's state evolves.* **Definition 5.2 (EPP Transition Function):** *The EPP Transition Function Π_{EPP} is a mapping:*

$$\Pi_{EPP} : (G, S_G, \mathcal{C}_{sys}, O_{trig}) \to S'_G$$

where $G = (V_G, E_G, \dots)$ is a CausaloidGraph, $S_G : V_G \to \{active, inactive\}$ is its current state, $\mathcal{C}_{sys}$ is the Context Collection, O_{trig} is the set of current input triggers, and $S'_G : V_G \to \{active, inactive\}$ is the updated state of the CausaloidGraph.

The computation of S'_G from S_G is achieved through an iterative evaluation process driven by the structure of G and the logic of its components):

1. ***Initialization:*** *Some nodes in V_G may have their states in S_G initially set or updated directly by O_{trig}.*

2. ***Iterative Evaluation & Propagation:*** *The process iteratively (or recursively, depending on the traversal strategy) considers nodes $v\_g \in V\_G$:*

    (a) *The activation of a node v_g (i.e., $S'_G(v_g)$) is determined by evaluating its causal function $f_{\chi_{v_g}}$. This evaluation takes into account:*
       - *Relevant observations from O_{trig} or the activation states of its source nodes (elements of V_{source} from incoming hyperedges e_g).*
       - *Contextual information obtained via ContextAccessor($\mathcal{C}_{refs_{\chi_{v_g}}}$).*

68

> (b) *The influence of source nodes V_{source} on target nodes V_{target} is mediated by the $logic_e$ of the connecting hyperedge $e_g = (V_{source}, V_{target}, logic_e)$. This $logic_e$ determines if and how the states of source nodes contribute to triggering the evaluation of target nodes.*

3. **Convergence:** *The process continues until no further changes in node states occur for the given O_{trig} and $\mathcal{C}_{sys}$, resulting in the stable updated state S'_G. For CausaloidGraphs with cycles, specific convergence criteria or iteration limits may be necessary.*

This operational definition emphasizes that effect propagation is a structured traversal and evaluation, where the "rules" of propagation are embedded in both the individual Causaloid functions f_χ and the connective logic $logic_e$ of the CausaloidGraph's hyperedges.

9.7.1   The Generative Function (Φ_{gen})

The EPP philosophy posits that the Contextual Fabric and even the Causal Structures themselves can be dynamic and emergent, shaped by an underlying "generative function." This formalism operationalizes this concept as Φ_{gen}, a function or set of functions that govern the evolution of $\mathcal{C}_{sys}$ and/or G.

Conceptual Role of Phi_gen in EPP

Φ_{gen} represents the meta-level rules or processes that can alter the EPP's structural components. It embodies the system's capacity for adaptation, learning, or emergence beyond simple state changes within a fixed structure. While the EPP framework itself does not mandate a specific physical interpretation for an ultimate generative function, it provides the means to model systems where such generative dynamics are at play operationally.

Generation and Dynamics of Contexts (Phi_gen_C)

Let Φ_{gen_C} be the component of Φ_{gen} responsible for the evolution of the Context Collection.

Definition 9.22 (Context Generative Function)**.**

$$\Phi_{gen\_C} : (\mathcal{C}_{sys}, O_{in}, E_{ext}) \to \mathcal{C}'_{sys}$$

where O_{in} represents relevant system inputs/observations and E_{ext} represents external events or triggers. $\mathcal{C}'_{sys}$ is the updated Context Collection. This function can:

- *Modify Contextoids $v \in V_C$ within a $C_i \in \mathcal{C}_{sys}$ via their adj_v protocol.*

- *Modify the structure of a Context Hypergraph C_i, e.g., by adding/removing Contextoids from V_{C_i} or Hyperedges from E_{C_i}.*

- *Add new Context Hypergraphs to $\mathcal{C}_{sys}$ or remove existing ones.*

The specific rules defining Φ_{gen_C} are domain-dependent but allow for contexts that structurally adapt to new information or changing environmental conditions (e.g., the dynamic temporal hypergraph described in the EPP philosophy [118]).

Generation and Dynamics of Causal Structures (Phi_gen_G)]

Let Φ_{gen_G} be the component of Φ_{gen} responsible for the evolution of CausaloidGraphs, enabling emergent causality.

Definition 9.23 (Causal Structure Generative Function)**.**

$$\Phi_{gen\_G} : (G, S_G, \mathcal{C}_{sys}, O_{in}, E_{ext}) \to (G', S'_G)$$

This function can modify the CausaloidGraph $G = (V_G, E_G, \dots)$ into a new graph $G' = (V'_G, E'_G, \dots)$ with a corresponding initial state S'_G. Modifications can include:

- *Adding or removing causal nodes v_g from V_G.*

- *Modifying the payload of existing nodes (e.g., changing a Causaloid's function f_χ or its linked assumptions $\mathcal{A}_{linked}$).*

- *Adding or removing causal hyperedges e_g from E_G, or altering their $logic_e$.*

This function formalizes the EPP's capacity for causal emergence, where the causal "rules of the game" themselves evolve in response to the system's experience and context.

Co-evolution of Context and Causal Structure (Phi_gen_Total)]

Definition 9.24 (Total Generative Function).

$$\Phi_{gen\_Total} : (G, S_G, \mathcal{C}_{sys}, O_{in}, E_{ext}) \to (G', S'_G, \mathcal{C}'_{sys})$$

This represents the combined action of Φ_{gen_C} and Φ_{gen_G}, allowing for feedback between structural changes in context and structural changes in causal models.

9.7.2 Core EPP Principles

The formalism presented supports and makes explicit the core philosophical principles of EPP.

Spacetime Agnosticism

The EPP formalism is inherently spacetime-agnostic. Neither the Contextoid (Def 9.2), Context Hypergraph (Def 9.9), Causaloid (Def 9.11), nor CausaloidGraph (Def 9.15) definitions mandate a spatiotemporal nature for the underlying fabric or relations. Spatial or temporal properties are introduced only if specific $payload_v$ types (Time, Space, Space-Time) are used within Contextoids. The core propagation function Π_{EPP} (Def 9.1) and generative function Φ_{gen} (Def 9.22) operate on these abstract structures; they only interact with spatiotemporal concepts if explicitly encoded within particular Causaloid functions f_χ or the rules of Φ_{gen}. This formally detaches EPP causality from a presupposed spacetime, fulfilling a key requirement for modeling non-Euclidean systems or systems where spacetime is emergent.

Emergence of Temporal Order

While EPP does not assume a universal linear temporal order, an operational or effective temporal order can emerge within the system through several mechanisms:

- **Contextual Sequencing:** Sequences of Time(t) Contextoids, or relations ($kind_e$) like "temporally_precedes" defined in E_C, can establish a local or domain-specific temporal order within $\mathcal{C}_{sys}$.
- **Propagation Dynamics:** The iterative application of Π_{EPP} inherently defines evaluation steps. While these steps are not necessarily universal time, they form a sequence of state transitions $S_G \to S'_G \to S''_G \ldots$ that constitutes an internal process time.
- **Generative Function Dynamics:** Changes orchestrated by Φ_{gen} also occur in sequence, creating a history of structural evolution.

Causaloid functions f_χ can then be designed to query and utilize this emergent temporal information from the context or process history when relevant.

Congruence with Classical Causality

Classical causality (A causes B if A precedes B and influences B) can be shown as a special case within the EPP formalism under specific conditions:

1. The Contextual Fabric $\mathcal{C}_{sys}$ is structured to represent a classical (possibly dynamic, as in GR) spacetime with a well-defined linear temporal order.
2. Causaloids χ_A and χ_B represent events A and B.
3. A causal hyperedge e_g connects χ_A to χ_B, with $logic_e$ such that χ_B is activated if χ_A is active AND the context (queried via f_{χ_B} or through $logic_e$) confirms that the event corresponding to χ_A occurred at an earlier time than the event corresponding to χ_B within the defined spacetime context.
4. The Causaloid χ_A can be designated "the cause" and χ_B "the effect" based on this emergent, contextually-defined temporal precedence.

Under these (and potentially other simplifying) assumptions, EPP's effect propagation aligns with classical cause-and-effect chains.

9.8 The Model and Verification Layer

The EPP primitives are composed into a verifiable **Model**, which encapsulates the core reasoning logic along with its operational assumptions.

9.8.1 Observation, Assumption, and Inference

These are empirical grounding primitives for a model.

- An **Observation** O is a tuple $(id_O, \text{val}_O, \text{effect}_O)$ representing an empirical measurement.
- An **Assumption** A is a tuple $(id_A, \text{desc}_A, f_A)$ where $f_A : \mathcal{P}(\Pi_E) \rightarrow \{\text{true, false}\}$ is a verifiable predicate about the operational conditions.
- An **Inference** I is a tuple representing a tested hypothesis about a causal link.

9.8.2 Model ($\mathcal{M}$)

Definition 9.25 (Model). *A Model $\mathcal{M}$ is a tuple:*

$$\mathcal{M} = (id_M, author_M, desc_M, \mathcal{A}_{set}, \chi_M, \mathcal{C}_M)$$

where $id_M, author_M, desc_M$ are metadata, $\mathcal{A}_{set}$ is a set of Assumptions upon which the model's validity depends, χ_M is the root Causaloid representing the model's core logic, and $\mathcal{C}_M$ is the Context it operates on.

9.9 The Causal State Machine (CSM)

The Effect Propagation Process (EPP) framework provides mechanisms for modeling how effects propagate and causal structures evolve. To translate the insights derived from such processes into concrete operations or interventions, the **Causal State Machine (CSM)** is introduced. The CSM serves as an orchestrator, linking specific causal conditions (represented as Causal States) to predefined deterministic Actions.

Let $\mathbb{I}_{CSM}$ be a set of unique identifiers for Causal States within a CSM.

Definition 9.26 (Causal State). *A **Causal State** q represents a specific condition whose truthfulness is determined by the evaluation of an associated Causaloid. It is defined as a tuple:*

$$q = (id_q, data_q, \chi_q, version_q)$$

where:

- *$id_q \in \mathbb{I}_{CSM}$ is a unique identifier for this Causal State.*

- *$data_q \in \mathcal{O}_{type}$ is the specific input observation or data value (compatible with the Causaloid's expected observation type $\mathcal{O}_{type}$) used for evaluating this state. This data may be intrinsic to the state or provided externally during evaluation.*

- *χ_q is a reference to a Causaloid (as defined in Section 9.6.1), $\chi_q = (id_{\chi_q}, f_{\chi_q}, \mathcal{C}_{refs_q}, \dots)$. The function f_{χ_q} embodies the predicate for this state.*

- *$version_q \in \mathbb{N}$ is an optional version number for the state definition.*

The activation of Causal State q is determined by $\text{eval}(q) = f_{\chi_q}(data_q, \text{ContextAccessor}(\mathcal{C}_{refs_q}))$.

Definition 9.27 (Causal Action). *A **Causal Action** a represents a deterministic operation. It is defined as a tuple:*

$$a = (exec_a, descr_a, version_a)$$

where:

- *$exec_a : void \rightarrow Result(void, ActionError)$ is the executable function representing the action. Its invocation may modify a WorldState implicitly.*

- *$descr_a$ is a descriptive label for the action.*

- *$version_a \in \mathbb{N}$ is an optional version number for the action definition.*

Definition 9.28 (Causal State Machine). *A Causal State Machine M is defined by its collection of Causal State-Action pairs. Let $\mathcal{Q}$ be the set of all possible Causal States and $\mathcal{A}$ be the set of all possible Causal Actions.*

$$M = (\mathcal{SA}_M)$$

where:

- $\mathcal{SA}_M \subseteq \{(q,a) \mid q \in \mathcal{Q}, a \in \mathcal{A}\}$ *is a finite set of ordered pairs, where each pair (q_j, a_j) associates a unique Causal State q_j (identified by id_{q_j}) with a Causal Action a_j. In implementation, this is often represented as a map from $id_q \rightarrow (q,a)$.*

The set of CausaloidGraphs $\mathcal{G}_M$ supervised by M is implicitly defined as the set of all CausaloidGraphs containing any Causaloid χ_q referenced by any q in $\mathcal{SA}_M$. Similarly, the set of relevant Context Hypergraphs $\mathcal{C}_M$ is the union of all $\mathcal{C}_{refs_q}$ for all q in $\mathcal{SA}_M$.

9.9.1 CSM Operation: State Evaluation and Action Triggering

The operation of the CSM involves evaluating its Causal States and triggering associated Causal Actions.

Single State Evaluation and Action (eval_single_state): Given a CSM $M = (\mathcal{SA}_M)$, an identifier $id_q^* \in \mathbb{I}_{CSM}$, and potentially new input data $d_{in} \in \mathcal{O}_{\text{type}}$ for that state:

1. Retrieve the state-action pair (q^*, a^*) from $\mathcal{SA}_M$ such that $id_{q^*} = id_q^*$. If no such pair exists, an error is indicated.
2. The Causal State $q^* = (id_{q^*}, \text{data}_{q^*}, \chi_{q^*}, \dots)$ is evaluated. The evaluation function is $f_{\chi_{q^*}}$. The input data used is either d_{in} (if provided externally for this evaluation) or the state's intrinsic data_{q^*}. Let trigger $= f_{\chi_{q^*}}(\text{data}_{\text{eval}}, \text{ContextAccessor}(\mathcal{C}_{refs_{q^*}}))$.
3. If trigger $=$ true, then the action a^* is executed by invoking its function $\text{exec}_{a^*}()$.

All States Evaluation and Action (eval_all_states): Given a CSM $M = (\mathcal{SA}_M)$:

1. For each state-action pair $(q_j, a_j) \in \mathcal{SA}_M$:
   (a) The Causal State $q_j = (id_{q_j}, \text{data}_{q_j}, \chi_{q_j}, \dots)$ is evaluated using its intrinsic data data_{q_j}. Let $\text{trigger}_j = f_{\chi_{q_j}}(\text{data}_{q_j}, \text{ContextAccessor}(\mathcal{C}_{refs_{a_j}}))$.
   (b) If $\text{trigger}_j =$ true, then the action a_j is executed by invoking its function $\text{exec}_{a_j}()$.

The order and concurrency of action execution in step 1(b) depend on the CSM's specific execution semantics (e.g., sequential, parallel, prioritized), which are an implementation detail beyond this core formalism.

9.9.2 CSM Dynamics: Managing State-Action Pairs

The set of state-action pairs $\mathcal{SA}_M$ within a CSM M can be dynamic, allowing the CSM to adapt its stimulus-response behavior. Formal operations on $\mathcal{SA}_M$ include:

- **AddStateAction**(M, q_{new}, a_{new}): $\mathcal{SA}'_M = \mathcal{SA}_M \cup \{(q_{new}, a_{new})\}$, provided $id_{q_{new}}$ is not already a key in $\mathcal{SA}_M$.
- **RemoveStateAction**(M, id_q): $\mathcal{SA}'_M = \mathcal{SA}_M \setminus \{(q,a) \mid id_q \text{ is the identifier of } q\}$.
- **UpdateStateAction**$(M, id_q, q_{updated}, a_{updated})$: Replaces the pair (q,a) associated with id_q with $(q_{updated}, a_{updated})$.

9.10 The Teleological Fabric ($\mathbb{E}$)

To ensure that a system's actions remain aligned with pre-defined goals, safety protocols, or ethical principles, the EPP introduces a teleological layer of governance. This is formalized as the **Effect Ethos**, a deontic reasoning engine that evaluates proposed actions from the Causal State Machine against a set of computable norms. This formalism is derived from the Defeasible Deontic Inheritance Calculus (DDIC) [7] and the DeepCausality reference implementation.

9.10.1 The Teloid (τ)

The fundamental, atomic unit of a computable norm is the **Teloid**. It encapsulates a single, verifiable rule.

Definition 9.29 (Teloid). *A Teloid τ is defined as a tuple:*

$$\tau = (id_\tau, a_{id}, p_\tau, m_\tau, t_\tau, s_\tau, prio_\tau, tags_\tau, meta_\tau)$$

where:

- $id_\tau \in \mathbb{I}$ *is a unique identifier for the Teloid.*

- a_{id} *is a string identifier for the class of action the norm governs.*

- $p_\tau : (\mathcal{C}_{sys}, A_p) \to \{true, false\}$ *is the **activation predicate**, a function that determines if the norm is active based on the system Context and a Proposed Action.*

- $m_\tau \in \{Impermissible, Obligatory, Optional(c)\}$ *is the **deontic modality**, where c is an associated integer cost for optional norms.*

- $t_\tau \in \mathbb{N}$ *is a timestamp for the Lex Posterior heuristic.*

- $s_\tau \in \mathbb{N}$ *is a specificity value for the Lex Specialis heuristic.*

- $prio_\tau \in \mathbb{N}$ *is a priority value for the Lex Superior heuristic.*

- $tags_\tau$ *is a set of string tags for filtering and indexing.*

- $meta_\tau$ *is an optional set of key-value metadata.*

9.10.2 Proposed Action (A_p) and Verdict (V)

The Effect Ethos evaluates a **Proposed Action** and returns a **Verdict**.

Definition 9.30 (Proposed Action). *A Proposed Action A_p is a tuple representing an action to be evaluated:*

$$A_p = (id_A, name_A, params_A)$$

where id_A is a unique identifier, $name_A$ is a descriptive name, and $params_A$ is a set of parameters for the action.

Definition 9.31 (Verdict). *A Verdict V is the outcome of the evaluation:*

$$V = (m_V, J_V)$$

where m_V is the final deontic modality, and J_V is the justification, a set of id_τ of the Teloids that led to the verdict.

9.10.3 The Effect Ethos Structure ($\mathbb{E}$)

The Effect Ethos engine, $\mathbb{E}$, comprises the norms and the relationships between them.

Definition 9.32 (Effect Ethos). *The Effect Ethos $\mathbb{E}$ is defined as a tuple:*

$$\mathbb{E} = (S_\tau, I_{tag}, \mathcal{G}_\tau)$$

where:

- S_τ *is the **Teloid Store**, a map from a Teloid ID to a Teloid: $S_\tau : id_\tau \mapsto \tau$.*

- I_{tag} *is the **Tag Index**, a map from a tag to a set of Teloid IDs: $I_{tag} : tag \mapsto \{id_{\tau_1}, id_{\tau_2}, \dots\}$.*

- $\mathcal{G}_\tau$ *is the **Teloid Graph**, a directed graph representing deontic relationships. It is a tuple (V_τ, E_τ), where V_τ is the set of all id_τ in the store, and E_τ is a set of directed edges $(id_{\tau,i}, id_{\tau,j}, r)$. The relation $r \in \{Inherits, Defeats\}$ defines how norms interact.*

9.10.4 The Deontic Inference Process ($\Pi_{Deontic}$)

The core dynamic of the Effect Ethos is the deontic inference process, $\Pi_{Deontic}$, which evaluates a proposed action.

Definition 9.33 ((Deontic Inference). *The inference process is a mapping:*

$$\Pi_{Deontic} : (\mathbb{E}, A_p, \mathcal{C}_{sys}, \textit{Tags}_{filter}) \rightarrow V$$

This process is executed in the following stages:

1. ***Filtering:*** *A candidate set of Teloids, τ_{cand}, is selected from the Teloid Store S_τ by retrieving all Teloid IDs associated with the input $\textit{Tags}_{filter}$ via the Tag Index I_{tag}.*

2. ***Activation:*** *A set of active Teloids, τ_{active}, is formed by evaluating the activation predicate p_τ for each $\tau \in \tau_{cand}$ against the Proposed Action A_p and the Context $\mathcal{C}_{sys}$.*

$$\tau_{active} = \{\tau \in \tau_{cand} \mid p_\tau(\mathcal{C}_{sys}, A_p) = \textit{true}\}$$

3. ***Conflict Resolution:*** *A resolution function, $f_{resolve}$, is applied to the active set τ_{active}. This function traverses the Teloid Graph $\mathcal{G}_\tau$ and queries the Normative Ontologies to apply the DDIC heuristics (Lex Posterior, Lex Specialis, Lex Superior). Using the timestamp, specificity, and priority attributes of the Teloids, it resolves conflicts defined by `Defeats` edges to yield a final, conflict-free set of justified norms, τ_{final}.*

4. ***Verdict Finding:*** *A consensus function, $f_{consensus}$, determines the final verdict V from τ_{final}. It applies a strict precedence order to the modalities: Impermissible > Obligatory > Optional. The justification J_V consists of the IDs of all Teloids in τ_{final}.*

9.11 Subsumption of Classical Causal Models

9.11.1 Mapping Pearl's Ladder of Causation

Judea Pearl's Ladder of Causation [4] provides a three-level hierarchy for causal reasoning. Each rung is mapped to distinct EPP mechanisms, demonstrating how the EPP's architecture provides a unified framework for association, intervention, and counterfactuals. Let G be a `CausaloidGraph` representing the Structural Causal Model (SCM).

Rung 1: Association (Seeing)

Concept   Association addresses queries of the form $P(Y|X)$, reasoning about the statistical relationship between variables based on passive observation.

EPP Formalization   In the EPP, association is the direct result of the `EffectPropagationProcess` (Π_{EPP}).

1. An observation, such as $X = x$, is represented as an initial `PropagatingEffect`, π_{initial}, which is part of the input trigger set, O_{trig}.

2. The inference is the evaluation of the `CausaloidGraph` G starting from the `Causaloid` χ_X that corresponds to the variable X. This is formally described by the EPP Transition Function (Definition 9.21):

$$\Pi_{EPP}(G, S_G, \mathcal{C}_{\text{factual}}, O_{trig}) \rightarrow S'_G$$

3. The resulting activation state of the `Causaloid` χ_Y within the updated state S'_G represents the association $P(Y|X)$.

Rung 2: Intervention (Doing)

Concept   Intervention addresses queries of the form $P(Y|\text{do}(X = x))$, predicting the effect of a deliberate action that forces a variable X to take on a value x.

EPP Formalization   Intervention is enacted by the `Causal State Machine (CSM)` (Section 9.9).

1. A `CausalState` q (Definition 9.26) is defined. Its activation is determined by its associated `Causaloid` χ_q, which evaluates whether a condition for intervention is met (e.g., based on an associative inference from Rung 1).

2. A `CausalAction` a (Definition 9.27) is defined. Its executable function, exec_a, modifies the `ContextualFabric` $\mathcal{C}$ to represent the intervention $\text{do}(X = x)$. This is typically achieved by updating a `Contextoid` v_x within the relevant `Context` C.

3. The CSM $M = (\mathcal{S}\mathcal{A}_M)$ contains the state-action pair (q, a). When the evaluation $\text{eval}(q)$ returns true, the CSM triggers the function $\text{exec}_a()$, formally executing the intervention on the system's context.

Rung 3: Counterfactuals (Imagining)

Concept   Counterfactuals address retrospective queries of the form $P(Y_{x'}|X = x, Z = z)$, imagining what the outcome Y would have been had X been x', given the factual observation that X was x and Z was z.

EPP Formalization   Counterfactuals are resolved via **Contextual Alternation**, a core feature of the EPP that leverages the separation of causal logic from its data context.

1. The factual observation $(X = x, Z = z)$ is represented by the state of `Contextoids` within a factual context, $\mathcal{C}_{\text{factual}}$.

2. A counterfactual context, $\mathcal{C}_{\text{counterfactual}}$, is created. This is typically a clone of $\mathcal{C}_{\text{factual}}$ where the `Contextoid` v_x is modified to reflect the hypothetical condition $X = x'$ using its adjustability protocol (e.g., $\text{update}_v(\text{payload}_{x'})$).

3. The counterfactual outcome is computed by evaluating the *same, unmodified* `CausaloidGraph` G against the newly created $\mathcal{C}_{\text{counterfactual}}$:
$$\Pi_{EPP}(G, S_G, \mathcal{C}_{\text{counterfactual}}, O_{trig}) \rightarrow S'_{G_{\text{cf}}}$$

4. The resulting activation state of the `Causaloid` χ_Y in the final state $S'_{G_{\text{cf}}}$ provides the answer to the counterfactual query.

9.11.2   Mapping Dynamic Bayesian Networks (DBN)

A Dynamic Bayesian Network (DBN) models a temporal process by representing dependencies between variables across discrete time slices. The EPP models a DBN by employing a static `CausaloidGraph` that is evaluated iteratively over a dynamic, temporal `Context`. This approach externalizes the temporal evolution from the causal logic into the contextual fabric.

Concept   A DBN defines the probability distribution of state variables at time t based on their state at $t - 1$, i.e., $P(V_t|V_{t-1})$.

EPP Formalization   The mapping of a DBN to EPP primitives is as follows:

1. **Time Slices and State Variables:** The timeline is represented within a dynamic `ContextHypergraph` $\mathcal{C}$. Each time point t corresponds to a `Tempoid` v_t. The state of a DBN variable V at time t is stored in a `Datoid` $v_{d,t}$, which is relationally linked to v_t within $\mathcal{C}$.

2. **Causal Logic:** Each DBN state variable V is represented by a single, reusable `Causaloid`, χ_V. The dependencies between variables within a single time slice (intra-slice dependencies) are modeled as a static `CausaloidGraph` G.

3. **Conditional Probability Tables (CPTs):** The CPT for a variable V, which defines $P(V_t|\text{Parents}(V_t))$, is encapsulated as the logic within the `causal_function` f_{χ_V} of the corresponding `Causaloid` χ_V.

4. **Temporal Dependencies:** Inter-slice dependencies (from time $t - 1$ to t) are not modeled as edges in the `CausaloidGraph` G. Instead, the `causal_function` f_{χ_V} for a `Causaloid` χ_V uses the `ContextAccessor` to query the `Context` $\mathcal{C}$ for the state of its parents at the previous time step, which it identifies via the `Tempoid` v_{t-1}.

5. **Inference (Filtering):** The DBN filtering process is modeled as an iterative loop. At each time step t:

   (a) The `EffectPropagationProcess` Π_{EPP} is executed on the static graph G, using the current context $\mathcal{C}_t$ to resolve temporal dependencies.

   (b) The results of the propagation are used to update the `Context` from $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$ via the `Context Generative Function` Φ_{gen_C}. This typically involves adding a new `Tempoid` v_{t+1} and its associated `Datoids`.

This methodology separates the static causal laws (the `CausaloidGraph`) from the evolving state of the world (the dynamic `Context`), providing a flexible framework for temporal causal modeling.

9.11.3 Mapping Granger Causality

Granger causality is a statistical concept of causality based on prediction. A time series X is said to Granger-cause another time series Y if past values of X contain information that helps predict future values of Y beyond the information contained in past values of Y alone. The EPP models this as a counterfactual query comparing prediction errors under different contextual conditions.

Concept Does the history of time series X improve the prediction of time series Y?

EPP Formalization The Granger causality test is formalized via **Contextual Alternation**.

1. **Time Series Data:** The historical data for both time series X and Y are represented as a sequence of `Datoids` within a `ContextHypergraph`, $\mathcal{C}_{\text{factual}}$. Each `Datoid` is linked to a `Tempoid` to preserve the temporal ordering.

2. **Predictive Model:** The predictive logic (e.g., a Vector Autoregression model) is encapsulated within the `causal_function` f_χ of a single, context-aware `Causaloid`, $\chi_{\text{predictor}}$. This function is designed to use the history of X if it is available in its context.

3. **The Test via Contextual Alternation:**
   (a) A **factual evaluation** is performed by executing $\chi_{\text{predictor}}$ against the complete factual context, $\mathcal{C}_{\text{factual}}$. The resulting prediction is π_{factual}.
   $$\pi_{\text{counterfactual}} = f_\chi(\pi_{in}, \text{ContextAccessor}(\{\mathcal{C}_{\text{counterfactual}}\}))$$
   (b) A **counterfactual context**, $\mathcal{C}_{\text{counterfactual}}$, is created from $\mathcal{C}_{\text{factual}}$ by removing all `Datoids` corresponding to the history of time series X.
   (c) A **counterfactual evaluation** is performed by executing the exact same `Causaloid` $\chi_{\text{predictor}}$ against the counterfactual context. The resulting prediction is $\pi_{\text{counterfactual}}$.
   $$\pi_{\text{counterfactual}} = f_\chi(\pi_{in}, \text{ContextAccessor}(\{\mathcal{C}_{\text{counterfactual}}\}))$$

4. **Conclusion:** Let y_{actual} be the true future value of Y. The prediction errors are compared. If the error of the factual prediction is significantly lower than the error of the counterfactual prediction, then X is said to Granger-cause Y.
   $$|\pi_{\text{factual}} - y_{\text{actual}}| < |\pi_{\text{counterfactual}} - y_{\text{actual}}| \implies \text{X Granger-causes Y}$$

9.11.4 Mapping Rubin Causal Model (RCM)

The Rubin Causal Model (RCM) defines a causal effect by comparing potential outcomes for a single unit under different treatment assignments. The EPP naturally models RCM's core concepts through its contextual alternation mechanism.

Concept The Individual Treatment Effect (ITE) for a unit i is defined as $Y_i(1) - Y_i(0)$, where $Y_i(1)$ is the outcome if unit i receives treatment, and $Y_i(0)$ is the outcome if unit i does not.

EPP Formalization The RCM is formalized in the EPP using **Contextual Alternation** to compute potential outcomes.

1. **Unit Representation:** An individual unit i is represented by a base `Context`, $\mathcal{C}_i$, containing its pre-treatment covariates as `Contextoids`.

2. **Causal Model:** The underlying causal laws that determine the outcome are represented by a `CausaloidGraph` G.

3. **Potential Outcomes via Contextual Alternation:**
   (a) A **treatment context** $\mathcal{C}_{i,\text{treatment}}$ is created by adding a `Datoid` $v_T = 1$ (representing treatment) to $\mathcal{C}_i$.
   (b) A **control context** $\mathcal{C}_{i,\text{control}}$ is created by adding a `Datoid` $v_T = 0$ (representing no treatment) to $\mathcal{C}_i$.

4. **Computation of Potential Outcomes:**
   (a) $Y_i(1)$ is the final `PropagatingEffect` $\Pi_{E,1}$ resulting from evaluating the `CausaloidGraph` G against the treatment context:
   $$\Pi_{EPP}(G, S_G, \mathcal{C}_{i,\text{treatment}}, O_{trig}) \rightarrow S'_{G,1}$$

(b) $Y_i(0)$ is the final `PropagatingEffect` $\Pi_{E,0}$ resulting from evaluating the `CausaloidGraph` G against the control context:

$$\Pi_{EPP}(G, S_G, \mathcal{C}_{i,\text{control}}, O_{trig}) \rightarrow S'_{G,0}$$

5. **Individual Treatment Effect (ITE):** The ITE for unit i is the computed difference between the final `PropagatingEffect`s:

$$\text{ITE}_i = \Pi_{E,1} - \Pi_{E,0}$$

This mapping directly operationalizes the RCM's counterfactual logic within the EPP by leveraging the explicit separation of causal mechanisms from contextual data.

9.11.5 Mapping Conditional Average Treatment Effects (CATE)

Conditional Average Treatment Effects (CATE) aim to estimate the average treatment effect for a specific subgroup of a population, conditioned on their characteristics. The EPP extends the RCM mapping to handle CATE by leveraging its ability to manage multiple contexts and aggregate results.

Concept   CATE is defined as $\tau(x) = E[Y(1) - Y(0)|X = x]$, representing the average treatment effect for individuals with covariates $X = x$.

EPP Formalization   CATE is formalized in the EPP by building upon the RCM mapping and extending it to handle subgroups.

1. **Subgroup Definition:** A subgroup of the population is represented as a set of factual `Context`s, $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$, where each $\mathcal{C}_i$ contains the covariates $X = x$ for a specific individual i.

2. **Learned Function:** The function $\tau(x)$, which estimates the CATE (e.g., derived from a machine learning model), is encapsulated as the `causal_function` f_χ within a `CausaloidGraph` G.

3. **Individual Treatment Effect (ITE) Calculation:** For each individual $\mathcal{C}_i$ in the subgroup, the ITE is calculated exactly as described in the RCM mapping (Section **??**). This involves:

   (a) Creating a treatment context $\mathcal{C}_{i,\text{treatment}}$ and a control context $\mathcal{C}_{i,\text{control}}$ for individual i.
   (b) Evaluating the `CausaloidGraph` G against both contexts to obtain $\Pi_{E,1}$ and $\Pi_{E,0}$.
   (c) Computing $\text{ITE}_i = \Pi_{E,1} - \Pi_{E,0}$.

4. **CATE Calculation:** The Conditional Average Treatment Effect for the subgroup is the statistical average of all computed ITE_i values:

$$\text{CATE}(x) = \frac{1}{n} \sum_{i=1}^{n} \text{ITE}_i$$

This mapping demonstrates how the EPP's modularity and contextual reasoning capabilities allow for the composition of complex causal inference tasks, such as CATE, from simpler primitives.

9.12 EPP Example: Smoking, Tar, and Cancer

To illustrate the core concepts of the Effect Propagation Process (EPP) formalism in a more concrete manner, we consider the well-known causal chain: Smoking → Tar in Lungs → Lung Cancer. This example demonstrates how Causaloids, Contexts (implicitly for this simplification), and a CausaloidGraph can represent this system, and how effect propagation leads to an inference. This example is simplified and inspired by the structure typically modeled as a Directed Acyclic Graph (DAG) in classical causal inference [4], here adapted to EPP.

9.12.1 Defining the Causal Problem

We aim to model a system to infer the likelihood of lung cancer based on smoking habits, mediated by tar accumulation. The core causal hypotheses are:

1. Smoking (represented by nicotine levels) leads to increased tar in the lungs.
2. Increased tar in the lungs leads to a higher risk of lung cancer.

We will represent these as individual Causaloids within a CausaloidGraph. For simplicity, any contextual dependencies (e.g., thresholds for "high" nicotine or tar) are assumed to be encapsulated within the Causaloid functions themselves or drawn from an implicitly defined context for this illustrative purpose. Observations will be represented as numerical values.

9.12.2 Formalizing the Components

The Causaloid (chi)

We define two primary Causaloids:

- $\chi_{S \to T}$ (**Smoking → Tar**):
    - $id_{\chi_{S \to T}} = 1$ (a unique identifier)
    - $\text{type}_{\chi_{S \to T}} = \text{Singleton}$
    - $f_{\chi_{S \to T}} : \mathcal{O}_{\text{nicotine}} \to \{\text{true}, \text{false}\}$, where $\mathcal{O}_{\text{nicotine}}$ is a numerical input representing nicotine level. The function $f_{\chi_{S \to T}}$ evaluates to true if the nicotine level exceeds a predefined threshold (e.g., 0.55), indicating a significant likelihood of tar presence due to smoking.
    - $\mathcal{C}_{refs} = \emptyset$ (assuming context-free logic for this example, or that thresholds are part of $f_{\chi_{S \to T}}$).
    - $\text{desc}_{\chi_{S \to T}}$ = "Causal relation between smoking and tar in the lung."
    - $\mathcal{A}_{linked}, I_{linked}$ = Optional, representing linkage to foundational assumptions or prior inferences regarding this relationship.

- $\chi_{T \to C}$ (**Tar → Cancer**):
    - $id_{\chi_{T \to C}} = 2$
    - $\text{type}_{\chi_{T \to C}} = \text{Singleton}$
    - $f_{\chi_{T \to C}} : \mathcal{O}_{\text{tar}} \to \{\text{true}, \text{false}\}$, where $\mathcal{O}_{\text{tar}}$ is a numerical input representing tar level. The function $f_{\chi_{T \to C}}$ evaluates to true if the tar level exceeds a predefined threshold (e.g., 0.55).
    - $\mathcal{C}_{refs} = \emptyset$.
    - $\text{desc}_{\chi_{T \to C}}$ = "Causal relation between tar in the lung and lung cancer."
    - $\mathcal{A}_{linked}, I_{linked}$ = Optional.

The shared underlying logic for f_χ in this example (input observation ≥ threshold) is abstracted into each Causaloid's specific function.

The CausaloidGraph (G)

We construct a CausaloidGraph $G_{STC} = (V_G, E_G, ID_G, \text{Name}_G)$ to represent the causal chain.

- $V_G = \{v_{g1}, v_{g2}\}$, where:
    - $v_{g1} = (id_{g1}, \text{payload}_{g1} = \chi_{S \to T})$
    - $v_{g2} = (id_{g2}, \text{payload}_{g2} = \chi_{T \to C})$
- $E_G = \{e_{g1}\}$, representing the link from smoking/tar to tar/cancer. For this simple chain:

- $e_{g1} = (V_{\text{source}} = \{v_{g1}\}, V_{\text{target}} = \{v_{g2}\}, \text{logic}_{e1})$
  - logic_{e1}: Specifies that the evaluation of v_{g2} is conditioned by, or follows, the evaluation of v_{g1}. In a simple direct propagation, if v_{g1} becomes active, this contributes to the conditions for evaluating v_{g2}. (In a computational system with distinct inputs for each step, this logic might simply define the sequence or dependency).

- ID_G, Name_G: Appropriate identifiers and names (e.g., "Smoking-Tar-Cancer Model").

This formal G_{STC} represents the structure. In a practical implementation, a collection of Causaloids (like a vector) reasoned over sequentially can instantiate such a linear graph.

9.12.3 Formalizing Effect Propagation (Π_{EPP})

Consider input observations $O_{trig} = \{(id_{\chi_{S \to T}}, o_{\text{nicotine}}), (id_{\chi_{T \to C}}, o_{\text{tar}})\}$, where o_{nicotine} is the observed nicotine level and o_{tar} is the observed tar level. Let the initial state be $S_G(v_{g1}) = \text{inactive}$, $S_G(v_{g2}) = \text{inactive}$.

The propagation $\Pi_{EPP}((G_{STC}, S_G, \emptyset, O_{trig})) \to S'_G$ conceptually proceeds as follows, reflecting a sequential evaluation for this chain:

1. **Evaluate $\chi_{S \to T}$ (node v_{g1}):** The function $f_{\chi_{S \to T}}$ is evaluated with its corresponding input o_{nicotine} from O_{trig}. If $f_{\chi_{S \to T}}(o_{\text{nicotine}}) = \text{true}$ (e.g., nicotine ≥ 0.55), then $S'_G(v_{g1}) = \text{active}$. Else, $S'_G(v_{g1}) = \text{inactive}$.

2. **Evaluate $\chi_{T \to C}$ (node v_{g2}):** The function $f_{\chi_{T \to C}}$ is evaluated with its corresponding input o_{tar} from O_{trig}. The evaluation of v_{g2} in this chained model might also be implicitly conditioned on v_{g1} being active if the overall inference requires the full chain to hold. If $f_{\chi_{T \to C}}(o_{\text{tar}}) = \text{true}$ (e.g., tar ≥ 0.55), then $S'_G(v_{g2}) = \text{active}$. Else, $S'_G(v_{g2}) = \text{inactive}$.

3. **Final Inference/System State:** The overall state of the system might be defined by a conjunction: Cancer risk is inferred if $S'_G(v_{g1}) = \text{active} \land S'_G(v_{g2}) = \text{active}$. This reflects whether the complete causal pathway from smoking, via tar, to cancer is deemed active for the given observations.

This illustrative propagation shows how individual Causaloid evaluations, based on specific inputs, contribute to the overall state of the causal model. The logic_e of the hyperedge e_{g1} here implies a dependency or sequential consideration in this chain.

9.12.4 Observations

This simplified example illustrates:

- **Causaloids** (χ) as operational units (Section 9.6.1), encapsulating specific causal functions (f_χ).

- A simple **CausaloidGraph** (G) structure (Section 9.6.3) representing the causal chain.

- The **Effect Propagation Process** (Π_{EPP}) (Section **??**) as the evaluation of Causaloids in a sequence determined by the graph structure and input data.

- The **State of the CausaloidGraph** (S_G) (Definition **??**) evolving based on these evaluations.

While this example omits explicit dynamic Contexts ($\mathcal{C}_{sys}$) or a dynamic Generative Function (Φ_{gen}) for brevity, it demonstrates how the core EPP structural entities map to a well-understood causal scenario.

9.13 Discussion

The preceding sections have laid out a formal, set-theoretic definition of the Effect Propagation Process (EPP). This formalism is a direct response to the need for new conceptual and computational tools to understand causality in systems where classical assumptions of a fixed spatiotemporal background and linear temporal order are demonstrably insufficient. The motivation stems from both the frontiers of fundamental physics, where spacetime itself is considered emergent, and the practical challenges of modeling complex adaptive systems across various scientific and engineering domains.

The primary significance of the EPP formalism lies in its principled detachment from spacetime. By making Context ($\mathcal{C}$) an explicit, definable, and potentially non-Euclidean, dynamic fabric, and by defining Causaloids (χ) as operational units of effect transfer independent of a priori temporal ordering, EPP achieves a level of generality that classical causal formalisms cannot. This enables the modeling of causality in non-physical or abstract domains, systems with complex multi-scale temporal dynamics and feedback loops, and, crucially, systems where causal relationships themselves can

emerge, transform, or dissolve in response to evolving contexts (dynamic regime shifts). The recursive isomorphism of CausaloidGraphs (G) further enhances expressive power, allowing for modular construction of complex causal models, while the clear separation of causal logic, contextual data, and propagation mechanisms facilitates transparency.

This foundational formalism, while potent, has current limitations. The EPP does not, in itself, provide algorithms for automated causal discovery of its structures from raw data; it provides the language to represent and reason with such structures once hypothesized. Compared to established frameworks like Pearl's SCMs, EPP offers greater flexibility for cyclic and emergent systems but currently lacks an equivalent to the extensive *do*-calculus.

10   The Implementation of the Effect Propagation Process

10.1   Overview

DeepCausality is an open-source framework, hosted at the Linux Foundation and accessible at `https://deepcausality.com`, designed to enable the construction, execution, and rigorous management of explicit, context-aware, and explainable causal models. It implements the Effect Propagation Process and with that provides a pathway to reason about cause and effect within intricate, multi-dimensional, and dynamically evolving environments. The framework's unique hypergeometric nature refers to its core reliance on hypergraph structures for representing both the rich tapestry of context and the complex web of causal relationships, offering a new level of expressiveness and analytical depth.

DeepCausality's core contributions are designed to provide a robust foundation for causally-grounded intelligence:

- **A Novel Hypergraph-based Context Engine:** At its heart, DeepCausality features a sophisticated engine for managing context. This moves beyond simple conditioning variables to enable the creation of intricate context hypergraphs populated by *Contextoids* – specialized nodes representing rich, multi-dimensional information encompassing Data, Time, Space, and Spacetime. This inherently supports dynamically adjustable contexts, the simultaneous integration of information from multiple distinct context hypergraphs (potentially with differing Euclidean or non-Euclidean geometries), and grounds causal reasoning in a highly nuanced and comprehensive understanding of the operational environment.

- **Structurally Composable Causal Modeling:** DeepCausality introduces *Causaloids* – encapsulated, testable causal functions – as the fundamental building blocks of causal models. These are organized within *CausaloidGraphs*, which are themselves hypergraphs explicitly representing intricate causal relationships. Crucially, this architecture employs recursive isomorphic causal data structures: nodes within a CausaloidGraph can themselves be entire subgraphs or collections of other causes. This enables the intuitive, modular construction of deeply complex, layered causal systems where macro-level phenomena can be decomposed into interacting micro-level mechanisms, ensuring transparent composability.

- **The Causal State Machine (CSM) for Actionable Intelligence:** Bridging the gap between causal understanding and effective intervention, the CSM is architected to manage interactions between causal models and their contexts. Based on the collective causal inference derived—the identification of specific active causes or system states—the CSM deterministically initiates predefined actions, facilitating the creation of complex, dynamic control and supervision systems that respond with causally-reasoned precision.

- **Implementation in Rust for Performance and Reliability:** Recognizing the demanding requirements of a sophisticated causal reasoning engine operating on potentially vast and dynamic data, DeepCausality is implemented in Rust. This choice leverages Rust's high-performance characteristics, memory safety guarantees, and expressive type system to build an efficient, robust, and reliable foundational causal engine.

This section details the complete architecture of DeepCausality, its conceptual foundations, and its Rust implementation.

10.2 UltraGraph

The importance of hypergraphs in the EPP led to the decision to implement a hypergraph in a dedicated Rust crate called UltraGraph. Initially, UltraGraph wrapped the MatrixGraph implementation of the Petgraph[10] crate. This was a deliberate decision to speed up the bootstrapping of the DeepCausality project while preserving the option to replace the implementation when the need arose. And indeed, as the requirements of the DeepCausality project kept advancing with the introduction of emergence, a new hypergraph implementation became necessary. The UltraGraph v0.8 release adds a ground-up rewrite of UltraGraph inspired by the NWHypergraph (NWHy)[119] architecture that was further optimized for Rust's memory model.

The key elements of the new UltraGraph implementation are:

- The introduction of a dual-state graph life-cycle.
- The introduction of a SoA CsrAdjacency type.
- The separation of forward and backward CsrAdjacency.

10.2.1 Dual-state graph life-cycle

UltraGraph v0.8 introduces a dual-state architecture. This recognizes that graph-based systems that implement the EPP have two distinct phases: a dynamic "Evolve" phase, where the structure is built and modified, and a stable "Analyze" phase, where high-speed queries are essential.

The Dynamic Graph State:

This is the default state for every new graph. It is an adjacency-list-based structure optimized for flexibility. Adding nodes and edges is a cheap O(1) operation, perfect for systems where the graph structure emerges dynamically over time.

The Static Graph State:

When the graph is ready, a one-time call to *.freeze()* transforms the graph into a hyper-optimized, immutable Compressed Sparse Row (CSR) format designed for peak performance. All algorithms check if a graph is in a frozen state; therefore, it is impossible by design to run any graph algorithm on a dynamic graph.

The graph life-cycle:

A Graph begins in a flexible DynamicGraph state, optimized for fast, O(1) mutations as the structure evolves. When the state has been finalized, and causal reasoning can begin, a single *.freeze()* call transforms the graph into a hyper-optimized, immutable CsmGraph based on a cache-friendly Struct of Arrays (SoA) memory layout. This step is the key to eliminating cache misses and unlocking near-linear scaling. If the graph needs to evolve further, simply call *.unfreeze()*. There is a memory trade-off in the state transition because, in each state, the memory usage is roughly $(e + v)$ whereas during a state transition, the memory usage temporarily peaks at $(e + v)^2$ and that warrants careful consideration when a graph grows large, i.e., beyond 1 billion nodes.

10.2.2 SoA CsrAdjacency Type

In conventional CSR-based graph representations (like NWHypergraph), adjacency information is typically packed together row-wise in a "single structure" per edge or neighbor, which is technically an Array of Structs (AoS) layout, or in simple terms, "rows of neighbors." UltraGraph, however, takes a different approach by introducing a CsrAdjacency<W> type that implements a Struct of Arrays (SoA) pattern:

Listing 1: UltraGraph: CsrAdjacency

```
1  #[derive(Default)]
2  pub(crate) struct CsrAdjacency<W> {
3      pub(crate) offsets: Vec<usize>,
4      pub(crate) targets: Vec<usize>,
5      pub(crate) weights: Vec<W>,
6  }
```

---

[10]https://docs.rs/petgraph/latest/petgraph

The CsrAdjacency layout divides each component of the adjacency data (offsets, targets, and weights) into separate, contiguous memory regions:

- offsets: Starting positions of each node's adjacency list.
- targets: The target node indices for each edge.
- weights: Edge weights (optional).

The Struct of Arrays design of the CsrAdjacency has two advantages:

Better cache utilization: When performing traversal or shortest path algorithms, the CPU can stream just the fields it needs (often offsets and targets) without pulling in unnecessary weight data and thus avoiding cache pollution and improving CPU data prefecht.

SIMD-friendliness: SoA layouts enable vectorized processing (e.g., with AVX) far more easily than Array of Structs (AoS). Also, a SoA layout is easier for the compiler to optimize and thus unlock meaningful performance gains without the need for complex optimization.

10.2.3 Separation of Forward and Backward CsrAdjacency

Moreover, UltraGraph uses two separate CsrAdjacency instances: one for successor or outbound edges and another for backward or inbound edges. This dual-CSR setup is more explicit and efficient than mixing directions within a single row layout because it reduces CPU cache pollution and thereby directly supports fast and efficient algorithm implementations. UltraGraph deliberately traded a bit more memory for better algorithm performance, as shown in the benchmarks.

Listing 2: UltraGraph: Forward and Backward CsrAdjacency

```
1  #[derive(Clone)]
2  pub struct CsmGraph<N, W>
3  where
4      N: Clone,
5      W: Clone + Default,
6  {
7      // Node payloads, indexed directly by 'usize'.
8      nodes: Vec<N>,
9      // CsrAdjacency structure for forward traversal (successors).
10     forward_edges: CsrAdjacency<W>,
11     // CSR structure for backward traversal (predecessors).
12     backward_edges: CsrAdjacency<W>,
13     // Index of the designated root node.
14     root_index: Option<usize>,
15 }
```

The backward node list is particularly useful in causality-based inference algorithms, where backtracking is often required, and is thus particularly well suited for DeepCausality. Memory usage remains low due to the combined effects of the Struct of Arrays layout and the clean separation between forward and backward adjacency. This design leads to a predictable, flat memory layout with minimal overhead:

- No per-node allocation overhead.
- No padding, no vtables, no boxed pointers.
- No HashMaps, linked lists, or other complex types.
- No indexing needed due to simple offset.
- When a node has no inbound or outbound nodes or weights, there is zero allocation, thus saving memory.

Memory fragmentation is largely prevented because of the freeze/unfreeze operation in the graph evolution life-cycle. Calling *.freeze()* compacts the structure, which removes any prior allocation gaps from the dynamic phase and thus results in a clean, continuous memory structure.

10.2.4 Discussion

This new implementation of UltraGraph based on its innovative two-stage graph life-cycle and its CPU-cache-friendly design enables causal graph reasoning at scale. While it is unlikely in practice to see causal graphs exceeding a million nodes, a context may grow large. UltraGraph confidently handles graphs up to a 100 million nodes on consumer-grade hardware and would only need a specialized high-memory server when the graph size is expected to exceed a billion nodes. At this scale, the underlying CSR implementation will most likely benefit from contemporary ReRAM-based[120] hardware accelerators specifically designed to accelerate hypergraphs with sparse representation. Furthermore, the two-stage graph life-cycle directly supports the implementation of emergence in DeepCausality because it cleanly separates the mutation from the analysis phase and thus enables the four-stage process outlined in the EPP Ontology.

10.3 DeepCausality

10.3.1 Overview

This chapter bridges the formal Effect Propagation Process (EPP) meta-calculus, as defined in Chapter 10, with the concrete architectural patterns of a real-world implementation. It introduces DeepCausality, the open-source reference implementation of the EPP, demonstrating how the abstract concepts of dynamic causality are realized in practice.

The choice of the Rust programming language for DeepCausality is a deliberate engineering decision. The language's compile-time guarantees of memory and thread safety, combined with its high performance and expressive modern type system, are important for building of robust, verifiable, and safety-critical systems using the EPP and its implementation DeepCausality.

10.3.2 Implementing the Context

The Context, formalized in Section 10.4, is a core innovation of the EPP. Its implementation in DeepCausality is designed to be both powerful and flexible, serving as the dynamic fabric upon which causal reasoning operates. The key engineering decisions are reflected in its primary data structure, Context<D, S, T, ST, SYM, VS, VT>.

The Context struct encapsulates a hypergraph provided by the ultragraph crate. This choice is fundamental, as hypergraphs can natively represent the n-ary relationships required for complex contextual models. The nodes within this graph are Contextoids, which are themselves a polymorphic enum, ContextoidType, capable of wrapping various kinds of data such as Datoid, Tempoid, or Spaceoid. This design allows the Context to store heterogeneous information without resorting to dynamic dispatch, preserving performance.

To facilitate efficient access, the Context maintains an internal HashMap<ContextoidId, usize> that maps a Contextoid's public, stable ID to its internal graph index. This provides O(1) lookup complexity, which is critical for performance in context-aware reasoning. Furthermore, the Context struct contains an Option<HashMap<u64, Extra-Context» field, which directly implements the mechanism for contextual counterfactuals (Section 5.13). This allows the system to manage multiple, parallel realities by creating and reasoning over alternate context graphs.

Listing 3: The DeepCausality Context Struct

```
1  pub struct Context<D, S, T, ST, SYM, VS, VT>
2  where
3      D: Datable + Clone,
4      S: Spatial<VS> + Clone,
5      T: Temporal<VT> + Clone,
6      ST: SpaceTemporal<VS, VT> + Clone,
7      SYM: Symbolic + Clone,
8      VS: Clone,
9      VT: Clone,
10 {
11     id: ContextId,
12     name: String,
13     base_context: UltraGraphWeighted<Contextoid<D, S, T, ST, SYM, VS, VT>, u64>,
14     id_to_index_map: HashMap<ContextoidId, usize>,
15     extra_contexts: Option<HashMap<u64, ExtraContext<...>>>,
16     // ... other fields for indexing
17 }
```

10.3.3 Implementing The Causaloid

The Causaloid, formalized in Section 9.6.1, is implemented as a struct that achieves polymorphism without dynamic dispatch by using an enum, CausaloidType, to determine whether it represents a Singleton, a Collection, or a Graph.

The core of its functionality lies in its *causal_fn* and *context_causal_fn* fields. These fields hold function pointers, making the Causaloid a holder for any object-level calculus. The specific function signature is defined by the type aliases CausalFn and ContextualCausalFn. This design allows any function that matches the signature to be embedded within a Causaloid, providing a powerful mechanism for composing different causal logics.

Listing 4: The Causal Function Type Alias

```
1  // For stateless reasoning
2  pub type CausalFn = fn(effect: &PropagatingEffect) -> Result<PropagatingEffect,
       CausalityError>;
3
4  // For context-aware reasoning
5  pub type ContextualCausalFn<D, S, T, ST, SYM, VS, VT> =
6      fn(
7          effect: &PropagatingEffect,
8          context: &Arc<Context<D, S, T, ST, SYM, VS, VT>>,
9      ) -> Result<PropagatingEffect, CausalityError>;
```

A key design feature is the causaloid's internal CausalType enum, which allows a single `Causaloid` struct to represent one of three distinct structural forms:

1. A `Singleton`: This represents an elementary cause, directly holding an optional causal function. This function can be context-free (type `CausalFn`) or context-aware (type `ContextualCausalDataFn`), with the latter taking an immutable reference to a `Context` object.

2. A `Collection`: This allows a `Causaloid` to encapsulate an entire collection (e.g., a `Vec`) of other `Causaloids`. The evaluation of such a "collection" `Causaloid` typically involves applying reasoning logic over its constituent members, facilitated by the `CausableReasoning` trait.

3. A `CausalGraph`: This enables a `Causaloid` to encapsulate an entire `CausaloidGraph` struct, thereby directly realizing the recursive isomorphic nature of the causal modeling engine.

Listing 5: The Causal CausaloidType Enum

```
1  #[derive(Debug, Clone, Copy, PartialEq, PartialOrd)]
2  pub enum CausaloidType {
3      Singleton,
4      Collection,
5      Graph,
6  }
```

The CausalVec and CausalGraph types are type aliases in Rust, defined as shown below:

Listing 6: The Causal Vec and Graph Type Alias

```
1  pub type CausalVec<D, S, T, ST, SYM, VS, VT> =
2   Vec<Causaloid<D, S, T, ST, SYM, VS, VT>>;
3
4  pub type CausalGraph<D, S, TM, ST, SYM, VS, VT> =
5      CausaloidGraph<Causaloid<D, S, TM, ST, SYM, VS, VT>>;
```

The Causaloid type itself is then defined as a Rust struct:

Listing 7: The Causaloid Type

```
1  #[derive(Clone)]
2  pub struct Causaloid<D, S, T, ST, SYM, VS, VT>
3  where
4      D: Datable + Clone,
5      S: Spatial<VS> + Clone,
6      T: Temporal<VT> + Clone,
7      ST: SpaceTemporal<VS, VT> + Clone,
8      SYM: Symbolic + Clone,
9      VS: Clone,
10     VT: Clone,
11 {
12     id: IdentificationValue,
13     causal_type: CausaloidType,
14     causal_fn: Option<CausalFn>,
15     context_causal_fn: Option<ContextualCausalFn<D, S, T, ST, SYM, VS, VT>>,
16     context: Option<Arc<Context<D, S, T, ST, SYM, VS, VT>>>,
17     effect: ArcRWLock<Option<PropagatingEffect>>,
18     causal_coll: Option<Arc<CausalVec<D, S, T, ST, SYM, VS, VT>>>,
19     causal_graph: Option<Arc<CausalGraph<D, S, T, ST, SYM, VS, VT>>>,
20     description: String,
21     ty: PhantomData<(VS, VT)>,
22 }
```

This enum-based design, combined with generic type parameters for context-related data types (D, S, T, ST, V), makes the `Causaloid` struct remarkably versatile. The last calculated effect of this causaloid. It's an Option because a causaloid may not have been evaluated yet. The PhantomData is neccessary because the generic parameters VS and VT are not bound and thus need a PhantomData binding to ensure the generic bound check by the Rust compiler. Various constructors for the Causaloid struct are provided to create `Causaloids` of each `CausalType`, optionally associating them with specific `Context` instances if their causal logic is context-dependent.

10.3.4 Implementing The Causaloid Collection

The CausaloidCollection, formalized in Section 10.5.2, is implemented within the Causaloid enum structure. A Causaloid of type Collection wraps a vector of other Causaloids, $Vec < Causaloid < ... >>$. This approach maintains the isomorphic nature of the causal primitive, where a collection is treated with the same interface as a singleton.

The reasoning over the collection is governed by the AggregateLogic enum, which specifies how the effects of the individual members are combined into a single outcome.

Listing 8: The Causaloid Type

```
1  pub enum AggregateLogic {
2      All,
3      Any,
4      None,
5      Some(usize),
6  }
```

The evaluation logic is implemented in the CausableCollectionReasoning trait. This trait provides methods such as evaluate_deterministic, evaluate_probabilistic, and evaluate_mixed, which take an AggregateLogic as an argument and apply the corresponding boolean or probabilistic aggregation to the collection's members. This provides a direct and verifiable implementation of the aggregation mechanisms required for manytoone causal relationships.

10.3.5 The Causaloid Graph

The CausaloidGraph, the structural composition primitive from Section 10.5.3, is implemented as the Causaloid-Graph<T> struct. This struct is a high-performance wrapper around ultragraph::UltraGraphWeighted, a graph library chosen for its memory efficiency and speed.

A key architectural feature inherited from ultragraph is the dual-state lifecycle. The graph exists in either a Dynamic state, which is optimized for fast, O(1) mutations like adding nodes and edges, or a Static state, which uses a hyper-optimized Compressed Sparse Row (CSR) format for peak traversal performance. The transition is managed by freeze() and unfreeze() methods. This design is critical for emergent systems, as it allows the causal structure to be modified efficiently during an "Evolve" phase and then locked for a fast "Analyze" phase.

Listing 9: The DeepCausality CausaloidGraph Struct

```
1  #[derive(Clone)]
2  pub struct CausaloidGraph<T>
3  where
4      T: Causable + PartialEq + Clone + Display,
5  {
6      id: IdentificationValue,
7      // CausalGraph<T> is an alias for UltraGraphWeighted<T, u64>
8      graph: CausalGraph<T>,
9  }
```

The CausableGraph trait defines the primary API for interacting with the graph. Crucially, the recursive isomorphism of the EPP is realized here: because a Causaloid can be of type Graph, an entire CausaloidGraph can be encapsulated within a single node of another, higher-level graph. This allows for the construction of deeply nested and modular causal models that remain conceptually simple and computationally efficient.

10.3.6 Implementing Causal Reasoning

Causal reasoning in DeepCausality is the operationalization of the Effect Propagation Process (Section 9.4). The core dynamic is implemented in the CausableGraphReasoning trait, specifically the evaluate_subgraph_from_cause method. This algorithm executes a breadthfirst traversal of the CausaloidGraph. The process follows these steps:

1. **Initialization**: The process begins at a specified start_index (typically the graph's root) with an initial_effect. A queue is initialized with the starting node and its effect, and a visited set is created to prevent cycles.

2. **Iterative Evaluation**: The algorithm dequeues a (node, effect) tuple and evaluates the Causaloid at that node using the incoming effect. The Causaloid's internal function, whether a simple CausalFn or a contextaware ContextualCausalFn, is executed.

3. **Contextual Query**: If the Causaloid's function is contextaware, it uses its ContextAccessor (an Arc<Context>) to query the Context for any necessary data, making the reasoning contextdependent. This directly implements the ContextAccessor from defintion 9.10.

4. **Propagation**: The output PropagatingEffect from the current node's evaluation becomes the input for all its children nodes, which are then enqueued for the next iteration.

5. **Adaptive Dispatch**: If a Causaloid returns a PropagatingEffect of type RelayTo(target_index, new_effect), the control flow is dynamically altered. The evaluation queue is cleared, and the traversal jumps to the target_index, using new_effect as its input. This directly implements the adaptive reasoning modality from definition 9.1.

6. **Convergence**: The process continues until the queue is empty, at which point the graph has reached a stable state for the given initial trigger.

7. **Return:** The last propagated effect is then returned as the final result of the reasoning chain.

10.3.7 Implementing The Effect Ethos

10.3.8 Implementing The Deontic Inference

10.3.9 Implementing The Causal State Machine

10.3.10 Discussion

10.4 Benchmarks

10.4.1 UltraGraph Benchmarks

All benchmarks were completed on a 2023 Macbook Pro with a M3 Max CPU.

Dynamic Graph Benchmarks:

The dynamic graph structure, when the graph is in an unfrozen state, is optimized for efficient mutation. The table below summarizes the performance characteristics of the key operations.

Table 5: Dynamic Graph Mutation Performance

Benchmark Name	Graph Size	Operation	Estimated Time (Median)	Outliers Detected
small_add_node	10	add_node	29.099 ns	14% (14 / 100)
medium_add_node	100	add_node	45.864 ns	12% (12 / 100)
large_add_node	1,000	add_node	39.293 ns	11% (11 / 100)
small_get_node	10	get_node	3.9417 ns	8% (8 / 100)
medium_get_node	100	get_node	3.9849 ns	2% (2 / 100)
large_get_node	1,000	get_node	3.9916 ns	7% (7 / 100)

Benchmark source code is available in the UltrGraph Github repository[11].

Static Graph Benchmarks:

The new architecture causes the largest and most significant performance gains for algorithms running over large graphs (100k or more nodes) because of its close alignment with contemporary hardware. By combining an instantaneous O(1) lookup with a perfectly linear scan over a node's neighbors, Ultragraph creates the ideal scenario for the CPU's prefetcher to easily anticipates a straight-line sprint through memory. The result becomes more notable the more data the prefetcher can load ahead of time, thus the disproportional performance gains on larger graphs.

Table 6: Memory Usage and Scaling Performance on a Linear Graph

Number of Nodes	Memory Usage	eval_subgraph	eval_path	eval_cause
100,000	55 MB	0.68 ms	0.57 ms	**5.4 ns**
1,000,000	350 MB	11.12 ms	6.95 ms	**5.5 ns**
10,000,000	3 GB	114 ms	85.80 ms	**5.6 ns**
100,000,000	32 GB	1.23 s	0.98 s	**5.5 ns**

Benchmark source code is available in the DeepCausality Github repository[12].

Observations:

Constant Time to get a single node: The benchmark evaluate_single_cause returns always takes about 5.5. ns regardless of whether the node lookup happens before or during the benchmark loop and regardless of whether blackbox is used or not. The time does not change with the size of the graph because the implementation of the underlying get_node is just two O(1) array lookup to find the index and than a straight redirect to a virtual memory address, which in this case, is close to the physical limit of the hardware memory architecture.

Near-Linear Scalability: Both the memory usage and the execution time for the subgraph and shortest_path tasks appear to scale in a roughly linear fashion with the number of nodes. A 10x increase in nodes results in a roughly 10x increase in time and memory.

All benchmarks are single-threaded. The performance shown in all benchmarks is from a single core. Initial experiments showed that for graphs up to 1 million nodes, the overhead of even highly-optimized parallel libraries like rayon resulted in a net performance loss of 30% or more compared to the single-threaded version. This is a testament to the extreme efficiency of the CSR layout when paired with modern CPU caches and prefetchers. The results suggest that meaningful gains from concurrency will only appear on very large graphs (likely 10M-50M nodes and above). However, this requires a concurrency model carefully designed to avoid the cache-invalidation issues common in work-stealing schedulers (used by rayon and Tokio).

---

[11]https://github.com/deepcausality-rs/deep_causality/tree/main/ultragraph/benches
[12]https://github.com/deepcausality-rs/deep_causality/tree/main/deep_causality/benches

10.5 Validation and Verification

10.5.1 Overview

Regulated industries such as avionics, robotics, or defense have to comply with strict regulatory requirements for software certification. Rust as a language provides a significant amount of safety relevant features that are difficult to achieve with C/C++ while still providing compatibility with existing code C/C++. Furthermore, the existence of a certified Rust compiler enables the validation and verification process of Rust systems in regulated industries. Build on this foundation, the DeepCausality project, while not certified, provides a plausible pathway towards certification.

10.5.2 Toolchain

Rust a a language was chose because of its memory safety and the availability of certified compilers from vendors such as Ferrocene[13] (ISO 26262 ASIL D, IEC 61508 SIL 3, IEC 62304 / Class C) HighTec EDV-Systeme[14] (ISO 26262 ASIL D), and AdaCore[15].

10.5.3 Hermetic Builds and Tests

Beyond the language-level assurances, DeepCausality implements a robust build and test infrastructure specifically designed for high-integrity environments. Regulatory standards like ISO 26262 (Automotive Safety), IEC 61508 (Functional Safety), DO-178C (Avionics Software), and IEC 62304 (Medical Device Software) demand that the software build process fulfils:

- Reproducibility: Given the same inputs, the system must always produce the same outputs.
- Traceability: Every output (including test results) must be traceable back to its exact inputs.
- Independence: Tests must be independent of each other. The failure of one component should not mask or cause the failure of another.
- Controlled Environment: The testing environment must be fully specified and controlled.

The entire DeepCausality mono-repository is built and tested using the Bazel build system. Bazel's core principle of running tests in fully isolated, hermetic sandboxes ensures that each test executes independently, free from external state or interference from other tests. Bazel enforces hermeticity by:

- Reproducible Inputs: Bazel requires all inputs (source code, compilers, libraries, tools) to be explicitly declared. This makes builds and tests fully reproducible.
- Isolation: Each test runs in its own isolated environment (a "sandbox"). This guarantees isolation and prevents side effects.
- Traceability: Each artifact in Bazel can be fully traced via the build graph. Bazel supports search and through the build graph which means one can quickly find all affected dependencies in case an issue has been found.

10.5.4 Design Principles

The DeepCausality project has been designed with regulatory requirements in mind to streamline the validation and verification process that precedes certification. Specifically, DeepCausality is built upon three core principles that establish a baseline for trustworthiness:

- Zero External Dependencies
- Zero Unsafe Code
- Zero Macros in Libraries

---

[13]https://ferrocene.dev/en
[14]https://hightec-rt.com/products/rust-development-platform
[15]https://www.adacore.com/gnatpro-rust

10.5.5 Zero External Dependencies

The DeepCausality codebase, including its custom tensor and random number generation primitives, compiles without external third-party dependencies by default. The decision to implement a custom CausalTensor as foundation for the various algorithms used in the DeepCausality project as well as the decision to implement a custom random number generation that underpins the Uncertain crate fundamentally enabled the zero dependency achievement. Among other benefits, this ensures that the Effect Ethos's deontic inference process operates with predictable and auditable determinism, which is critical for proofing end to end alignment. The zero external dependency principle reduces supply chain risk by eliminating unknown code from external sources and effective mitigates a key security concern in high-integrity systems. It ensures that the entire intellectual supply chain is auditable and simplifies cross-platform compilation for diverse embedded hardware targets.

The only exception is an optional feature flag, os-random, in the deep_causality_rand crate. When enabled, this flag introduces a dependency on a thin Rust wrapper for libc to access the operating system's secure random number generator. This is clearly documented and is strictly opt-in for developers who require a cryptographically secure random numbers provided by the host system. Alternatively, the deep_causality_rand crate also contains a flexible trait that exposes the random number generator (RNG) functionality. For regulated industries, the trait-based architecture for RNG allows them to seamlessly swap in their own certified hardware RNG binding by simply implementing one trait, which means DeepCausality becomes compatible with existing certified hardware.

For the development process, there are a handful of external dependencies i.e. Criterion for running benchmarks or CSV for reading data files, but these are very limited, all contained outside the DeepCausality codebase, and only applicable to the development process with the understanding that none of these will be part of a production build.

10.5.6 Zero Unsafe Code

The entire DeepCausality codebase including all internal dependencies, examples, and tests, are implemented entirely in safe Rust. This preserves Rust's guarantee of memory safety and therefore eliminating an entire class of catastrophic failures (e.g., buffer overflows, data races) at compile time. It significantly reduces the burden of memory safety audits, which are typically one of the most complex and costly aspects of high-integrity software.

10.5.7 Zero Macros in Libraries

The entire DeepCausality codebase is free of macros to ensures maximum code clarity and audibility. The entire codebase can be read, understood, and analyzed line-by-line, which is essential for formal verification processes and human review in regulated environments. The only area where macros had to be used is in bulk testing for generic traits as, for example, in the deep_causality_num trait. The reason is quite obvious, core numerical traits such as "mul" that defines the multiplication operation for every numerical type would require identical tests for every possible numerical type in Rust. In those cases, a macro is used to derive the test mainly to keep the test files largely free of duplicates. It is important to understand that this practice is contained only to tests and only applies to very specific foundational crates i.e. deep_causality_rand and deep_causality_num.

These key principles are complemented by a rigorous testing regimen that maintains a 3-month rolling average code coverage of 97% across its 80,000+ lines of Rust code. While code coverage does not constitute formal verification, it is a critical quantitative indicator of testing thoroughness. For certification bodies, this high coverage, coupled with the inherent safety of Rust and the absence of external dependencies, substantially lowers the time, effort and cost associated with verification and validation activities. Combined, these three key principles enable a number of properties particular relevant to regulated industries:

- Supply Chain Security: By being entirely self-contained, DeepCausality guarantees the integrity of its code.
- Cross-Compilation & Embedded Systems: The std-only approach makes DeepCausality inherently portable across a vast array of embedded target platforms supported by Rust.
- Performance & Development Velocity: Because the DeepCausality codebase is self-contained, the entire mono-repository (est. 80K Lines of Code), completes a full clean build within 1o seconds translates directly to developer productivity.

The DeepCausality project provides a clear and plausible pathway towards regulatory certification because of the existence of a certified Rust compiler, the already established hermetic build system, and the achieved zero dependencies, unsafe, and macros best practice across the entire project codebase. Systems build with DeepCausality that seek certification still need to comply with all regulatory requirements, but because of the careful preparations already made, the process of validation and verification becomes manageable for an experienced team.

11 Future Work

The preceding chapters have established the Effect Propagation Process as a foundational framework for modeling dynamic contextual causality. The framework's philosophical underpinnings have informed its implementation in DeepCausality. However, because of the single axiomatic definition of generalized causality the EPP is built upon, the resulting vast space of new possibilities will inform future work that falls outside of the scope of this monograph.

11.1 Functional Causality

Based on the function theoretical conjecture established in section **??**, future work may explore to apply concepts from category theory like monads and functors to define operations such as map, flatmap or filter for causal function. By doing so, scholars of formal methods can formulate a Causal Calculus with the same level of rigor and expressive power as the Lambda Calculus and thus pave the way for causal programming.

Furthermore, future work may explore the synergy between causal functions and functional programming with an extension to define a type based causal effect system relative to an explicit context to use the vast existing work on type and effect theory to make causal models increasingly statically verifiable. Instead of relying on a simple type *PropagatingEffect*, a more expressive type *PropagatingEffect<T, C>* is known to apply only to type T on Context C. The compiler could then statically prove that a propagating effect is applied to the correct type and, critically, to the correct context thus preventing an entire category of incorrect model transfer errors at compile time thus directly supporting verifiable safety.

Functional programming methods, with their emphasis on immutability, pure functions, and referential transparency, inherently facilitate formal verification which simplifies validation and verification for causal models written in a functional causal programming language. A a functional causal programming language combined with the aforementioned type based causal effect system amount to a principled functional causality programming model rooted in function theory. The impact of functional causality programming on regulated industries can be substantial in terms of reduced validation effort and reduced certification time and cost.

11.2 Contextual Causal Discovery

A primary focus for future work is the development of a robust contextual causal discovery algorithm to automatically construct Causaloid Graphs for a given domain. The main challenge is rooted in the interaction between context and the causal model, and that results in a triple challenge.

First, for a given domain, one context graph structure may work better than others for a given model, thus finding an "optimal" context graph with respect to supporting the causal model is the first major challenge.

Second, when holding the context graph constant, the structure of the graph has a significant impact on the overall efficacy of the model and thus finding an optimal causal graph for a given context is the second major challenge. Unfortunately, the context and causal graph cannot be separated from each other; therefore, it is most likely that these two challenges are deeply interrelated and require an integrated approach that searches for context representation and the causal model in tandem.

Third, the causal function embedded into each causaloid also implies a nontrivial search, construction, and validation challenge. This challenge deeply interrelates with contemporary work on decomposing causality during the discovery process. However, one complication is that causaloids are isomorphically recursive and may contain another causaloid, a causal collection, or another causaloid graph to represent arbitrarily complex causal structures. It is not known to the author if sufficiently advanced algorithms that can discover complex nested causal structures exist.

Addressing the challenges imposed by contextual causal learning would require a hybrid approach. For example, for finding causal relations in raw data, an adaptation of the SURD algorithm[121] may result in a workable foundation. For the challenge of the entangled context and causal graph structure, algorithms from the field of search-based procedural content generation[122] may offer a path forward. Dealing with the expected computational cost resulting from the high-dimensional nature of the problem of finding two interdependent hypergraphs suggests the exploration of deep neuroevolution adapted to high dimensional data[123]. Furthermore, the broader exploration of novelty search[124] algorithms may yield deep insights about robust contextual causal discovery.

11.3 Teleological Inquiry

An intelligent agent must be proactive, capable of navigating a complex future to achieve its goals. One critical prerequisite to autonomous decision making is teleological inquiry. This would require a formalization of a calculus

to establish the foundation for verifiable teleological reasoning that enables the design and validation of systems whose adherence to specified intent or regulations is mathematically provable. From there, a teleological inquiry becomes feasible.

For a teleological inquiry, the outcome aims for alignment with the Effect Ethos, is held constant as the goal, whereas the decision space is a set of potential pathways through alternative contexts. The challenge is to decide which alternative course of action leads to an outcome that is most closely aligned with the encoded Effect Ethos. This requires a form of path-seeking and alignment inquiry. However, the classical formulation of inquiry as a search problem, as seen, for example in Zetetic Logic[125], is fundamentally incompatible with the EPP's real-time operational capabilities. Instead, the EPP proposes a multi-step combination of alternative contexts with qualitative diversity as foundation for an efficient teleological inquiry.

1) Candidate Action Generation:

At a decision point, the system generates a finite set of immediate, mutually exclusive Causal Actions it could take. For a drone, this could be an alternative trajectory. The existing context informs a heuristic that limits the number of potential causal actions to allow only actions that are feasible i.e., within the remaining power budget.

2) The Deontic Filter:

Before any computationally expensive simulation is performed, every candidate action is passed through a Deontic Filter. This filter applies only the subset of Teloids that represent absolute, non-negotiable rules encoded as impermissible norms. Any action that violates a hard deontic rule is immediately discarded and is not simulated. This "fail-fast" step prunes the action space, ensuring that computational resources are only spent on options that are, at a minimum, not forbidden.

3) Parallel Counterfactual Simulation:

The EPP's instantiates a temporary, hypothetical Context for each candidate action and runs a fast, forward simulation to predict the future state that would result if the proposed action were taken. This creates N predicted alternative futures.

4) Unified Scoring:

For each of the simulated "alternative futures," the EPP now calculates a single, unified Teleological Alignment Score (TAS). This score is a holistic measure of "quality" that seamlessly blends two criteria:

Goal Achievement: Positive scores for making progress towards the primary teleological objective i.e., reaching a destination coordinate.

Ethic/Safety score: Each outcome is also scored relative to adherence to ethics and safety rules encoded in the effect ethos. Positive scores represent relative adherence to the ethical boundaries. Negative scores representing the "cost" of violating Teloids (e.g., penalties for high fuel consumption, excessive engine strain, or entering a restricted airspace).

The TAS represents the quantitative expression of the Effect Ethos for each specific potential future.

5) Update the Tactical Option Map:

Instead of selecting any one particular set of proposed actions, the EPP maintains a persistent, low-dimensional grid called the Tactical Option Map. Each cell in this grid represents a region of the diverse behavioral space, for example, low energy options that sorts all actions by the remaining energy score, or a time to target options sort all actions by the lowest time to target regardless of the remaining energy.

At this stage, the system possesses an entire map of the best-known way to achieve different types of behaviors that supports vastly more sophisticated decision making processes:

- **Default selection:** Assuming no anomalies have been detected, the EPP can select the action with the highest overall TAS from the entire on the Tactical Option Map.
- **Contingency Planning:** If the highest scoring TAS option somehow becomes untenable i.e. due to changed visibility i.e. a drone is flying through a tunnel, the EPP can instantly search the Tactical Option Map for viable alternatives without complex re-planning.
- **Strategic Flexibility:** Because of the Tactical Option Map, priorities can change at at time and be quickly implemented by switching to different alternative in the Tactical Option Map.

The proposed teleological inquiry builds entirely on techniques that are either rooted in the EPP i.e., alternative contexts and implemented in DeepCausality or in proven algorithms such as qualitative diversity that have been used in the industry for close to two decades. When implemented, the potential for (semi) autonomous agents, however, can be substantial. As an example, a search and rescue drone may operate under a default directive of low energy to thermal scan the largest possible area for survivors in a disaster area. It is possible that another drone has detected multiple potential survivors, but requires support to scan the surrounding area in parallel with multiple other drones. In this case, the command center can override the priority directive with "fastest time" to dispatch one or more drones to the new target area and provide rapid expansion of thermal scans in the designated area of interest. Here, the teleological inquiry re-plans ad-hoc for the new priority, adjust the trajectory, and dispatches to the new target coordinates using the new trajectory calculated based on the new priority. One step further, the same teleological inquiry can also be used to re-configure the core mission of the drone. For example, suppose a thermal signature indicates multiple survivors have been identified with a visual confirms multiple injuries. Then, the drone can determine to pause thermal scan, descend, drop off a pre-packed first aid kit with a positioning beacon that guides the rescue crew thus saving valuable time. Following the drop, the drone may proceed with the thermal scan.

11.4 Indefinite Causal Structures

The frontier of modern physics, particularly in the pursuit of a unified theory of quantum gravity, has introduced the profound notion of Indefinite Causal Structures (ICS), wherein the temporal ordering of events may exist in a quantum superposition. While Indefinite Causal Structures originate in fundamental physics, its implications for computation and the modeling of complex systems are significant. However, a full formal treatment of quantum indefinite causal structures is beyond the scope of this foundational monograph, but the EPP provides a uniquely suitable foundation for extending the domain of computational causality to explore indefinite causal structures.

The EPP's foundational compatibility with ICS results from its core axiomatic principles. The EPP's principled detachment from a presupposed spacetime removes the rigid constraint of temporal precedence that has historically served as the bedrock for causal reasoning. As ICS directly refutes this fixed "happen-before" relation, the EPP has already established the necessary foundation to consider a reality where A causes B and B causes A can coexist simultaneously within a single process. The EPP's single axiom of causality as functional dependency, $E2 = f(E1)$, is deliberately unconstrained, allowing the causal function f to be instantiated as a process matrix, which is the precise mathematical formalism required to describe applied indefinite causal processes such as the quantum switch.

A prospective implementation of indefinite causal structures in the EPP would involve a series of targeted modifications. First, the PropagatingEffect requires a new variant that represents a probability distribution over a set of potential outcomes. Consequently, the CausaloidGraph itself would require an extension to its formal definition, enabling it to represent a superposition of causal orderings. This means, the superposition of a causal graph is defined as a set of different graphs with different causal connectivity patterns to allow for the simultaneous co-existence of diametrically different causal orderings. Then, the EPP itself needs to be generalized to a process capable of evaluating all pathways within the superposition set of all causal graphs, aggregating their outcomes to compute a final, coherent probability distribution for the final state of indefinite causal structure.

When implemented, the potential payoff extends into critical domains of engineering and computational science. It would provide the first comprehensive tool for the unified, multi-paradigm simulation of quantum computers and communication networks, where classical control systems modeled with standard EPP modalities could interact with an indefinite causal structure and devise interactions via the causal state machine. In fields such as drug discovery and molecular dynamics, where the precise sequence of binding events is governed by quantum-level causality, an ICS-enabled EPP could offer a more faithful modeling of reality, potentially accelerating the design of novel molecules. Similarly, in quantum communication, the classical control system could be modeled using standard EPP, while the quantum core and its interconnections are modeled using the new ICS extensions, creating a unified, multi-paradigm simulation environment.

Extending the EPP to incorporate indefinite causal structures promises to yield a powerful new class of computational tools for modeling the complex, dynamic, and fundamentally quantum realities for a new category of dynamic quantum causal systems.

11.5 Discussion

The EPP offers a deep, wide, and rich selection of multiple areas of future work that ranges from the integration of advanced quantum concepts like indefinite causal structure, covers new methods for causal learning, and expands computational ethics with the proposed teleological inquiry. The exploration of Indefinite Causal Structures seeks to formalize the generation of outcomes from a superposition of causal realities. The pursuit of Contextual Causal

Discovery aims to generate the causal model itself from the intricate interplay of causal logic and its context. The teleological inquiry seeks to guide the generation of future actions toward desirable and verifiable ends.

The presented areas of future work remain necessarily incomplete, as the full spectrum of possibilities opened by the EPP still has to be discovered. Each presented avenue of future work presents formidable technical and scientific challenges that imposes a demand for careful consideration of the epistemological and ethical consequences of engineering systems capable of dynamic self-modification. While the EPP provides the tools for this exploration, the responsibility, however, for its principled application remains an endeavor for the scientific and engineering communities.

12 Epilogue

The Effect Propagation Process, as detailed in the preceding sections, provides a consistent, first-principles foundation for dynamic causality and a new philosophy-grounded language to study dynamic, emergent, and contextual causality. It is important to acknowledge that the higher-order effects resulting from the single axiomatic foundation most likely will remain incomplete for the foreseeable future despite the EPPś attempt to describe and formalize an initial set. The vast space of new possibilities opened up by an irreducible single axiomatic definition will require some meaningful time to explore and understand. The presented three modalities of dynamic causality already demonstrated very clearly that the higher-order consequences need to be carefully considered to ensure safe and reliable dynamic systems built upon the EPP.

The implementation of the EPP in the DeepCausality[16] demonstrates the practicality of hypergeometric computational causality for fast context-aware causal reasoning across Euclidean and non-Euclidean spaces. Furthermore, Deep-Causality, as a reference implementation, provides an excellent foundation to study dynamic causality further. And indeed, the new foundation of dynamic causality already suggests several domains of inquiry:

- **The Theory of Hybrid Causal Models:** The EPP already enables uniform causal reasoning over deterministic and probabilistic modes and the Causaloid is a polymorphic container that can host SCMs, Bayesian networks, deterministic rules, and neural networks. One area the EPP opens up is the exploration of a theory of hybrid causal models to explore hybrid causal composition and discovery.
- **Formal verification of the CSM:** The Causal State Machine (CSM) bridges insights from causal reasoning to actions and that opens up the area of formal verification of the CSM especially for dynamic causal models with a mixture of static and dynamic contexts.
- **The Formalization of Causal Emergence:** While the EPP introduced causal emergence, its foundation is far from settled; therefore, the study of causal emergence promises further discoveries.
- **The Dynamics of Context:** The EPP established external context and its modalities, but especially the intersection between a dynamically changing environment and a dynamic context offers a promising area for study. This is particularly valuable for control systems in autonomous unmanned vehicles that have to adapt to new terrain.
- **The Teleology of Emergence:** Directly related to the calculus of purpose follows the study of teleological boundaries of causal emergence which is particularly relevant to the development of safe emergent algorithms.

The presented EPP serves as an open invitation. It is an invitation to the architects and engineers of complex systems who require formal proof of safety and explainability. It is an invitation to the underwriters of risk, who must quantify, model, and price complex risks that are contextual and dynamic. And it is an invitation to the regulators and policymakers tasked with ensuring the safe and compliant integration of complex systems into the fabric of society. Trust is the single most valuable asset any society has and the EPP exists to facilitate trust building and certification of complex dynamic causal systems.

---

[16]https://deepcausality.com

Appendix A: Formal Proof of SCM Subsumption

Overview

This proof demonstrates that the EPP is a strict generalization of the SCM, with the SCM being a special case of the EPP under specific constraints. This is achieved by constructing a mapping function from the set of all SCMs to the set of all EPP models and proving that this mapping preserves the core semantics of observation and intervention. By implication, any system representable by a Structural Causal Model (SCM) can be fully represented by an equivalent Effect Propagation Process (EPP) model.

Formal Definitions

We begin by formally defining the source and target structures of our mapping.

Definition 12.1 (Structural Causal Model (SCM)). *A Structural Causal Model $\mathcal{M}_{SCM}$ is a tuple $\langle \mathcal{U}, \mathcal{V}, \mathcal{F} \rangle$, where:*

- $\mathcal{U}$ *is a set of exogenous variables, which are uncaused within the model.*

- $\mathcal{V}$ *is a set of endogenous variables, $\{V_1, V_2, \ldots, V_n\}$.*

- $\mathcal{F}$ *is a set of structural functions, $\{f_1, f_2, \ldots, f_n\}$, such that each $V_i \in \mathcal{V}$ is determined by a function of its causal parents $PA_i \subseteq \mathcal{V}$ and a unique exogenous variable $U_i \in \mathcal{U}$:*

$$V_i = f_i(PA_i, U_i)$$

Definition 12.2 (EPP Model). *An EPP Model $\mathcal{M}_{EPP}$, as defined in Section 9.8, is a tuple $\langle G, \mathcal{C}, CSM \rangle$, where:*

- G *is a* `CausaloidGraph`.

- $\mathcal{C}$ *is a* `ContextualFabric`.

- CSM *is a* `CausalStateMachine`.

The Constructive Mapping Function (Φ)

We define a constructive mapping function $\Phi : \text{SCM} \to \text{EPP}$ that transforms any SCM $\mathcal{M}_{SCM}$ into a corresponding EPP Model $\mathcal{M}_{EPP}$.

1. **Map Exogenous Variables to Context:** For each exogenous variable $u_i \in \mathcal{U}$, we construct a corresponding `Contextoid` v_i within a single `ContextHypergraph` $\mathcal{C}_0$ in the `ContextualFabric` $\mathcal{C}$. The `payload` of v_i holds the value of u_i.
$$\Phi(\mathcal{U}) \mapsto \mathcal{C} \quad \text{where } \mathcal{C} = \{\mathcal{C}_0\} \text{ and } V_{\mathcal{C}_0} = \{v_i \mid u_i \in \mathcal{U}\}$$

2. **Map Endogenous Variables and Functions to Causaloids:** For each endogenous variable $v_i \in \mathcal{V}$ and its associated structural function f_i, we construct a `Singleton Causaloid` χ_i in the `CausaloidGraph` G.
$$\Phi(\mathcal{V}, \mathcal{F}) \mapsto G \quad \text{where } V_G = \{\chi_i \mid v_i \in \mathcal{V}\}$$

   The `causal_function` f_{χ_i} of each χ_i is defined to be a context-aware function that:
   (a) Accepts `PropagatingEffects` from its parent Causaloids (representing the values of PA_i).
   (b) Uses its `ContextAccessor` to query $\mathcal{C}_0$ for the value of the `Contextoid` v_i (representing u_i).
   (c) Computes and returns the value $f_i(PA_i, U_i)$ as its output `PropagatingEffect`.

3. **Map Causal Dependencies to Hyperedges:** For each dependency $V_i = f_i(PA_i, U_i)$, we construct a `CausalHyperedge` $e_i \in E_G$.
   - V_{source} of e_i is the set of Causaloids $\{\chi_j \mid V_j \in PA_i\}$.
   - V_{target} of e_i is the singleton set $\{\chi_i\}$.
   - logic_e is a simple propagation trigger, activating χ_i once all its parents are active.

This construction $\Phi(\mathcal{M}_{SCM})$ yields a complete EPP Model $\mathcal{M}_{EPP}$.

Proof of Semantic Equivalence

We now prove that the semantics of observation and intervention are preserved under the mapping Φ.

Case 1: Observation (Seeing)

In SCM The computation of $P(Y = y | X = x)$ is performed by setting the value of variable X to x, propagating values through the equations in $\mathcal{F}$ in topological order, and reading the resulting value of Y.

In EPP This corresponds to triggering the Effect Propagation Process Π_{EPP} on the graph G with an initial `PropagatingEffect` π_{initial} (carrying the value x) applied to the Causaloid χ_X. The propagation through the graph, dictated by the hyperedges, follows the same topological order as the SCM's directed acyclic graph. Because each f_{χ_i} is defined to be identical to f_i, the final `PropagatingEffect` produced by Causaloid χ_Y will be a distribution centered on the value of Y. Thus, the observational semantics are preserved.

Case 2: Intervention (Doing)

In SCM The intervention $\text{do}(X = x)$ is a "graph surgery" where the structural equation for X is replaced by the constant $X = x$, severing the influence of PA_X.

In EPP We model this using the `Causal State Machine (CSM)` and **Contextual Alternation**. We define a `CausalAction` $a_{\text{do}(x)}$ whose exec_a function modifies the context. It performs a stateful update on a designated "interventional context" $\mathcal{C}_{\text{int}}$, setting the value of a `Contextoid` v_x^{int} to x. The `causal_function` of the Causaloid χ_X is defined with a priority logic: it first checks its `ContextAccessor` for an interventional value in $\mathcal{C}_{\text{int}}$. If present, it returns that value directly, ignoring any inputs from its parent Causaloids. Otherwise, it proceeds with its standard computation.

Equivalence Executing the `CausalAction` $a_{\text{do}(x)}$ via the CSM ensures that any subsequent evaluation of G operates in a state where the interventional context is active. When the propagation reaches χ_X, its function finds the value x in $\mathcal{C}_{\text{int}}$ and returns it, thereby ignoring the `PropagatingEffect`s from its parents. This perfectly replicates the "graph surgery" of the *do*-operator by severing the functional dependency at runtime through a contextual override. Thus, the interventional semantics are preserved.

Conclusion

We have constructed a function Φ that provides a complete mapping from any SCM to an EPP model. We have demonstrated that this mapping preserves the core semantics of both observation and intervention. Therefore, any SCM can be expressed as a specialized instance of the EPP.The SCM corresponds to an EPP model that is constrained to a static `CausaloidGraph` and a static `ContextualFabric`.

The Effect Propagation Process (EPP): An Axiomatic Foundation for Dynamic Causality

Glossary

This glossary provides definitions for the key formal terms and symbols used throughout this paper to describe the Effect Propagation Process (EPP) and its components.

Effect Propagation Process (EPP)
> The overarching philosophical framework and the formalized dynamic process (Π_{EPP}) of effect transfer within a CausaloidGraph, conditioned by the Contextual Fabric.

Contextual Fabric ($\mathcal{C}$)
> The structured environment, composed of Context Hypergraphs, within which effects propagate and causal relationships are conditioned. It is formalized as a Context Collection $\mathcal{C}_{sys}$.

Contextoid (v)
> The atomic unit of contextual information, defined as a tuple $v = (id_v, \text{payload}_v, \text{adj}_v)$. It encapsulates data, time, space, or spacetime values. (See Definition 3.1)

Contextoid Payload (payload$_v$)
> The tagged union representing the actual data, temporal, spatial, or spatiotemporal value held by a Contextoid.

Context Hypergraph (C)
> A structured collection of Contextoids (V_C) and the N-ary relationships (Hyperedges E_C) between them, defined as $C = (V_C, E_C, ID_C, \text{Name}_C)$. (See Definition 3.2)

Context Collection ($\mathcal{C}_{sys}$)
> A finite set of distinct Context Hypergraphs, $\mathcal{C}_{sys} = \{C_1, C_2, \ldots, C_k\}$, representing the total available contextual information. (See Definition 3.4)

Context Accessor (ContextAccessor($\mathcal{C}_{refs}$))
> A functional interface providing read-access for Causaloids to a specified subset of Context Hypergraphs ($\mathcal{C}_{refs} \subseteq \mathcal{C}_{sys}$). (See Definition 3.5)

Causaloid (χ)
> The fundamental, operational unit of causal interaction, defined as $\chi = (id_\chi, \text{type}_\chi, f_\chi, \mathcal{C}_{refs}, \text{desc}_\chi, \mathcal{A}_{linked}, I_{linked})$. It encapsulates a testable mechanism for effect transfer. (See Definition 4.1)

Causaloid Type (type$_\chi$)
> Specifies the structural nature of a Causaloid (Singleton, Collection, or Graph), determining how its causal function f_χ is realized.

Causal Function (f_χ)
> The operational logic associated with a Causaloid χ, which maps input observations ($\mathcal{O}_{\text{type}}$) and accessed context to an activation status ({true, false}). (See Section 4.2.2)

CausaloidGraph (G)
> A hypergraph, defined as $G = (V_G, E_G, ID_G, \text{Name}_G)$, whose nodes ($V_G$) typically encapsulate Causaloids and whose hyperedges (E_G) define pathways and logic for effect propagation. Supports recursive isomorphism. (See Definition 4.2)

Causal Node (v_g)
> A node within a CausaloidGraph G, defined as $v_g = (id_g, \text{payload}_g)$, where payload$_g$ can be a Causaloid, a collection of Causaloids, or another CausaloidGraph.

Causal Hyperedge (e_g)
> A hyperedge within a CausaloidGraph G, defined as $e_g = (V_{\text{source}}, V_{\text{target}}, \text{logic}_e)$, representing a directed functional relationship.

State of CausaloidGraph (S_G)
> A function $S_G : V_G \to \{\text{active}, \text{inactive}\}$ mapping each causal node in a CausaloidGraph to its current activation status. (See Definition 4.3)

Effect (ε)
> Primarily the activation state (active/inactive) of a Causaloid within EPP, and the transfer of this status or derived information. (See Section 5.1.1)

Input Observations/Triggers (O_{input}, O_{trig})
> External data or events (O_{input} being the set of all possible, O_{trig} being the current set) that can initiate or influence the Effect Propagation Process. (See Definition 5.1)

EPP Transition Function (Π_{EPP})

The core dynamic function $\Pi_{EPP} : (G, S_G, \mathcal{C}_{sys}, O_{trig}) \to S'_G$, describing how the state of a Causaloid-Graph evolves. (See Definition 5.2)

Operational Generative Function (Φ_{gen})

A formal function or set of functions representing meta-level rules for the dynamic evolution of the Contextual Fabric (Φ_{gen_C}), CausaloidGraphs (Φ_{gen_G}), or both (Φ_{gen_Total}). (See Section 5.2)

Causal State Machine (CSM, M)

An operational component, defined by its set of state-action pairs $\mathcal{SA}_M$, that links recognized Causal States (q) to deterministic Causal Actions (a). (See Definition 6.3)

Causal State (q)

A specific condition within a CSM, $q = (id_q, \text{data}_q, \chi_q, \text{version}_q)$, whose activation is determined by its associated Causaloid χ_q. (See Definition 6.1)

Causal Action (a)

A deterministic operation within a CSM, $a = (\text{exec}_a, \text{descr}_a, \text{version}_a)$, triggered by an active Causal State. (See Definition 6.2)

References

[1] Alfred North Whitehead. *Process and reality*. Simon and Schuster, 2010.

[2] Henri Bergson, Donald Landes, and Elizabeth Grosz. *Creative evolution*. Routledge, 2022.

[3] Einstein Relatively Easy. Nov 1915 The four Einstein's papers. Website. Accessed on May 18, 2025.

[4] J Pearl. Causality: models, reasoning, and inference cambridge university press new york. *NY, USA*, 2000.

[5] Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv preprint arXiv:1801.04016*, 2018.

[6] Elias Bareinboim and Judea Pearl. Transportability of causal effects: Completeness results. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 698–704, 2012.

[7] Taylor Olson, Roberto Salas-Damian, and Kenneth D. Forbus. A defeasible deontic calculus for resolving norm conflicts, 2024.

[8] Bertrand Russell. On the notion of cause. Proceedings of the Aristotelian Society, Vol. 13, 1912. Pages 1–26.

[9] Lucien Hardy. Probability theories with dynamic causal structure: a new framework for quantum gravity. arXiv preprint, 2005. arXiv:gr-qc/0509120.

[10] Lucien Hardy. Probability theories with dynamic causal structure: a new framework for quantum gravity. *arXiv preprint gr-qc/0509120*, 2005.

[11] James Bornholt, Todd Mytkowicz, Kathryn S McKinley, and Kathryn McKinley. Uncertain< t >: A first-order type for uncertain data. March 2014.

[12] Richard Dacre Archer-Hind et al. *The Timaeus of Plato*. Macmillan, 1888.

[13] Martin Heidegger. *Aristotle's Metaphysics 1–3: On the Essence and Actuality of Force*. Indiana University Press, 1995.

[14] Andrea Falcon. Aristotle on causality. 2006.

[15] Lucius Annaeus Seneca. Seneca: Selected philosophical letters: Translated with introduction and commentary. Book, OUP Oxford, 2007.

[16] Stanford Encyclopedia of Philosophy. Leibniz's Philosophy of Physics. Stanford Encyclopedia of Philosophy, Online Resource. Accessed on May 18, 2025.

[17] John Stuart Mill. *A System Of Logic Ratiocinative And Inductive Vol. I*. BoD–Books on Demand, 2023.

[18] David Hume. *A treatise of human nature*. Oxford University Press, 2000.

[19] Immanuel Kant. *Critique of pure reason*, volume 6. Minerva Heritage Press, 2024.

[20] Hans Reichenbach. *The direction of time*, volume 65. Univ of California Press, 1991.

[21] Karl Popper. *The logic of scientific discovery*. Routledge, 2005.

[22] Luke Mrini and Lucien Hardy. Indefinite causal structure and causal inequalities with time-symmetry, 2024.

[23] Thomas S Verma. Causal networks: semantics and expressiveness. *Technical Report R-65, Cognitive Systems Laborator, University of California, Los Angeles*, 1986.

[24] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.

[25] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

[26] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[27] Judea Pearl. *Causality*. Cambridge university press, 2009.

[28] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT press, 2017.

[29] Juan Correa and Elias Bareinboim. A calculus for stochastic interventions: Causal effect identification and surrogate experiments. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 10093–10100, 2020.

[30] D. Janzing, R. Chaves, and B. Schölkopf. Algorithmic indepen-dence of initial condition and dynamical law in thermodynamics and causal inference. *New Journal of Physics*, 18(9), 2016.

[31] Jonas Peters, Stefan Bauer, and Niklas Pfister. Causal models for dynamical systems. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pages 671–690. ACM Digital Library, 2022.

[32] Karthika Mohan and Judea Pearl. Graphical models for processing missing data. *Journal of the American Statistical Association*, 116(534):1023–1037, 2021.

[33] Thomas S Richardson and Peter Spirtes. Causal inference via ancestral graph models. *Oxford Statistical Science Series*, pages 83–105, 2003.

[34] Thomas S. Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 1990.

[35] Eli Sherman, David Arbour, and Ilya Shpitser. General identification of dynamic treatment regimes under interference. In *International Conference on Artificial Intelligence and Statistics*, pages 3917–3927. PMLR, 2020.

[36] A. Balke and J. Pearl. Probabilistic evaluation of counterfactual queries. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, volume I, pages 230–237, Menlo Park, CA, 1994. MIT Press.

[37] S.L. Morgan and C. Winship. *Counterfactuals and Causal Inference: Methods and Principles for Social Research*. Analytical Methods for Social Research. Cambridge University Press, New York, NY, 2nd edition, 2015.

[38] Jin Tian and Judea Pearl. Probabilities of causation: Bounds and identification. *Annals of Mathematics and Artificial Intelligence*, 28(1):287–313, 2000.

[39] Jin Tian and Judea Pearl. On the identification of causal effects. *Technical Report R-290-L, Department of Computer Science, University of California, Los Angeles*, 2002.

[40] Ilya Shpitser, Thomas S Richardson, and James M Robins. An efficient algorithm for computing interventional distributions in latent variable causal models. *arXiv preprint arXiv:1202.3763*, 2012.

[41] Finnian Lattimore, Tor Lattimore, and Mark D Reid. Causal bandits: Learning good interventions via causal inference. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[42] Sanghack Lee and Elias Bareinboim. Structural causal bandits: Where to intervene? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[43] Jingwen Zhang, Yifang Chen, and Amandeep Singh. Causal bandits: Online decision-making in endogenous settings. arXiv preprint arXiv:2211.08649, 2022.

[44] Blair Bilodeau, Linbo Wang, and Daniel M. Roy. Adaptively exploiting d-separators with causal bandits, 2022.

[45] Jeroen Berrevoets, Sam Verboven, and Wouter Verbeke. Model-agnostic counterfactual reasoning for outcomes in dynamic environments. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 792–802, Jul 2021.

[46] Jeroen Berrevoets, Sam Verboven, and Wouter Verbeke. Treatment effect optimisation in dynamic environments. *Journal of Causal Inference*, 10(1):106–122, 2022.

[47] Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960, 1986.

[48] Jerzy Splawa-Neyman, Dorota M Dabrowska, and Terrence P Speed. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, pages 465–472, 1990.

[49] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

[50] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[51] Joshua D Angrist, Guido W Imbens, and Donald B Rubin. Identification of causal effects using instrumental variables. *Journal of the American statistical Association*, 91(434):444–455, 1996.

[52] Brantly Callaway and Pedro HC Sant'Anna. Difference-in-differences with multiple time periods. *Journal of econometrics*, 225(2):200–230, 2021.

[53] Guido W Imbens and Thomas Lemieux. Regression discontinuity designs: A guide to practice. *Journal of econometrics*, 142(2):615–635, 2008.

[54] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):947–1012, 2016.

[55] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[56] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International conference on machine learning*, pages 5815–5826. PMLR, 2021.

[57] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

[58] Frederick Eberhardt. Introduction to the foundations of causal discovery. *International Journal of Data Science and Analytics*, 3(2):81–91, 2017.

[59] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

[60] Uzma Hasan and Md Osman Gani. Kcrl: A prior knowledge based causal discovery framework with reinforcement learning. In *Machine Learning for Healthcare Conference*, pages 691–714. PMLR, 2022.

[61] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[62] Uzma Hasan, Emam Hossain, and Md Osman Gani. A survey on causal discovery methods for i.i.d. and time series data, 2024.

[63] Jing Ma, Mengting Wan, Longqi Yang, Jundong Li, Brent Hecht, and Jaime Teevan. Learning causal effects on hypergraphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1202–1212, 2022.

[64] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[65] Clive WJ Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and control*, 2:329–352, 1980.

[66] John Geweke. Measurement of linear dependence and feedback between multiple time series. *Journal of the American statistical association*, 77(378):304–313, 1982.

[67] P Padav. Granger causality in time series–explained using chicken and egg problem, 2021.

[68] Qin Zhang. Dynamic uncertain causality graph for knowledge representation and reasoning: Discrete dag cases. *Journal of Computer Science and Technology*, 27(1):1–23, 2012.

[69] H. C. Deng and Q. Zhang. Cubic DUCG (dynamic uncertain causality graph) recursive algorithm and its implementation in nuclear power plant fault diagnoses. M.S. thesis, Dept. Eng. Phys., Tsinghua Univ., Beijing, China, 2018.

[70] J. Hu, L. Zhang, A. Wang, and S. Li. Accident prevention by fault propagation analysis and causal fault diagnosis based on Granger causality test. In *Proc. 13th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, pages 1554–1558, Jul 2017.

[71] N. Deng and Q. Zhang. Towards Dynamic Uncertain Causality Graphs for the Intelligent Diagnosis and Treatment of Hepatitis B. *Symmetry*, 12:1690, 2020.

[72] Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):19, 2019.

[73] Licheng Jiao, Yuhan Wang, Xu Liu, Lingling Li, Fang Liu, Wenping Ma, Yuwei Guo, Puhua Chen, Shuyuan Yang, and Biao Hou. Causal inference meets deep learning: A comprehensive survey. *Research*, 7:0467, 2024.

[74] Jeroen Berrevoets, Krzysztof Kacprzyk, Zhaozhi Qian, and Mihaela van der Schaar. Navigating causal deep learning, 2022.

[75] Jeroen Berrevoets, Krzysztof Kacprzyk, Zhaozhi Qian, and Mihaela van der Schaar. Causal deep learning, 2024.

[76] Eli N Weinstein and David M Blei. Hierarchical causal models. *arXiv preprint arXiv:2401.05330*, 2024.

[77] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017. Extract from GRAPH HYPERNETWORKS.pdf [72].

[78] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[79] Xavier Ouvrard. HYPERGRAPHS: AN INTRODUCTION AND REVIEW. *arXiv preprint arXiv:2002.05014*, March 2020.

[80] Claude Berge. *Graphs and hypergraphs*, volume 7. North-Holland publishing company Amsterdam, 1973.

[81] Bilal Alsallakh, Luana Micallef, Wolfgang Aigner, Helwig Hauser, Silvia Miksch, and Peter Rodgers. The State-of-the-Art of Set Visualization: The State-of-the-Art of Set Visualization. *Computer Graphics Forum*, 35(1):234–260, February 2016.

[82] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLoS ONE*, 9(6):e98679, June 2014.

[83] U.V. Catalyurek and C. Aykanat. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 10(7):673–693, July 1999.

[84] K.D. Devine, E.G. Boman, R.T. Heaphy, R.H. Bisseling, and U.V. Catalyurek. Parallel hypergraph partitioning for scientific computing. In *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, page 10 pp., Rhodes Island, Greece, 2006. IEEE.

[85] Wenyin Yang, Guojun Wang, Md Zakirul Alam Bhuiyan, and Kim-Kwang Raymond Choo. Hypergraph partitioning for social networks based on information entropy modularity. *Journal of Network and Computer Applications*, 86:59–71, May 2017.

[86] Xiaoyao Zheng, Yonglong Luo, Liping Sun, Xintao Ding, and Ji Zhang. A novel social network hybrid recommender system based on hypergraph topologic structure. *World Wide Web*, 21(4):985–1013, July 2018.

[87] Denny Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in neural information processing systems*, pages 1601–1608, 2007.

[88] Wenhui Wu, Sam Kwong, Yu Zhou, Yuheng Jia, and Wei Gao. Nonnegative matrix factorization with mixed hyper-graph regularization for community detection. *Information Sciences*, 435:263–281, April 2018.

[89] Taisong Jin, Jun Yu, Jane You, Kun Zeng, Cuihua Li, and Zhengtao Yu. Low-rank matrix factorization with multiple Hypergraph regularizer. *Pattern Recognition*, 48(3):1011–1022, March 2015.

[90] Lei Zhu, Jialie Shen, Hai Jin, Ran Zheng, and Liang Xie. Content-Based Visual Landmark Search via Multimodal Hypergraph Learning. *IEEE Transactions on Cybernetics*, 45(12):2756–2769, December 2015.

[91] Yu Zhu, Ziyu Guan, Shulong Tan, Haifeng Liu, Deng Cai, and Xiaofei He. Heterogeneous hypergraph embedding for document recommendation. *Neurocomputing*, 216:150–162, December 2016.

[92] Vamsi K Potluru, Robert E Tillman, Prashant Reddy, and Manuela Veloso. Deeplex: A gnn for link prediction in multiplex networks. In *SIAM Workshop on Network Science*, 2020.

[93] M. Zhang and Y. Chen. Link Prediction Based on Graph Neural Networks. In *Proc. Advances in Neural Information Process-ing Systems*, 2018.

[94] Zizhen Deng, Xiaolong Zheng, Hu Tian, and Daniel Dajun Zeng. Deep causal learning: representation, discovery and inference. *arXiv preprint arXiv:2211.03374*, 2022.

[95] Vikas Ramachandra. Deep learning for causal inference. *arXiv preprint arXiv:1803.00149*, 2018.

[96] Takeshi Teshima and Masashi Sugiyama. Incorporating causal graphical prior knowledge into predictive modeling via simple data augmentation. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 86–96, Jul 2021.

[97] Sai Srinivas Kancheti, Abbavaram Gowtham Reddy, Vineeth N Balasubramanian, and Amit Sharma. Matching learned causal effects of neural networks with domain priors. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10676–10696, Jul 2022.

[98] Yangyi Lu, Amirhossein Meisami, Ambuj Tewari, and William Yan. Regret analysis of bandit problems with causal background knowledge. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 141–150, Aug 2020.

[99] Guy Tennenholtz, Uri Shalit, Shie Mannor, and Yonathan Efroni. Bandits with partially observable confounded data. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 430–439, Jul 2021.

[100] Beatrice Acciaio, Anastasis Kratsios, and Gudmund Pammer. Designing universal causal deep learning models: The geometric (hyper) transformer. *Mathematical Finance*, 34(2):671–735, 2024.

[101] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[102] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations, 2019.

[103] Amit Sharma and Emre Kiciman. Dowhy: An end-to-end library for causal inference. *arXiv preprint arXiv:2011.04216*, 2020.

[104] Miruna Oprescu, Vasilis Syrgkanis, Keith Battocchi, Maggie Hei, and Greg Lewis. Econml: A machine learning library for estimating heterogeneous treatment effects. In *33rd Conference on Neural Information Processing Systems*, page 6, 2019.

[105] Yang Zhao and Qing Liu. Causal ml: Python package for causal inference machine learning. *SoftwareX*, 21:101294, 2023.

[106] Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2023.

[107] Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. Can large language models infer causation from correlation? *arXiv preprint arXiv:2306.05836*, 2023.

[108] Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. Cladder: Assessing causal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:31038–31065, 2023.

[109] Zeyu Wang. Causalbench: A comprehensive benchmark for evaluating causal reasoning capabilities of large language models. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 143–151, 2024.

[110] Moritz Miller, Bernhard Schölkopf, and Siyuan Guo. Counterfactual reasoning: an analysis of in-context emergence, 2025.

[111] Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-pfn: In-context learning for causal effect estimation, 2025.

[112] Markus Kalisch and Peter Bühlman. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(3), 2007.

[113] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters, 2018.

[114] Defense Advanced Research Projects Agency (DARPA). Assured neuro symbolic learning and reasoning (ansr) summary. https://www.darpa.mil/research/programs/assured-neuro-symbolic-learning-and-reasoning. Accessed: May 12, 2025.

[115] Luciano Floridi. Applied philosophy of ai as conceptual design. *A Companion to Applied Philosophy of AI*, pages 31–38, 2025.

[116] Lucio M. Milanese, Nuno F. Loureiro, and Stanislav Boldyrev. Dynamic phase alignment in navier-stokes turbulence. *Physical Review Letters*, 127(27), December 2021.

[117] Montgomery Furth et al. *Metaphysics: Zeta, Eta, Theta, Iota. Books VII-X*. Hackett Publishing, 1985.

[118] Marvin Hansen. The effect propagation process: A philosophical framework for post-quantum causality. https://github.com/deepcausality-rs/papers, June 2025. Accessed: June 2025.

[119] Xu T Liu, Jesun Firoz, Assefaw H Gebremedhin, and Andrew Lumsdaine. Nwhy: A framework for hypergraph analytics: Representations, data structures, and algorithms. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 275–284. IEEE, 2022.

[120] Long Zheng, Ao Hu, Qinggang Wang, Yu Huang, Haoqin Huang, Pengcheng Yao, Shuyi Xiong, Xiaofei Liao, and Hai Jin. Phgraph: A high-performance reram-based accelerator for hypergraph applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(5):1318–1331, 2023.

[121] Álvaro Martínez-Sánchez, Gonzalo Arranz, and Adrián Lozano-Durán. Decomposing causality into its synergistic, unique, and redundant components. *Nature Communications*, 15(1), November 2024.

[122] Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186, 2011.

[123] Cédric Colas, Vashisht Madhavan, Joost Huizinga, and Jeff Clune. Scaling map-elites to deep neuroevolution. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 67–75, 2020.

[124] Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.

[125] Jared Millson. A defeasible calculus for zetetic agents. *Logic and Logical Philosophy*, 30(1):3–37, October 2020.