THE EFFECT PROPAGATION PROCESS (EPP): A DYNAMIC THEORY OF CAUSALITY

Marvin Hansen
marvin.hansen@gmail.com
Date: July 23, 2025

ABSTRACT

Contents

1 Introduction

The study about cause and effect has served mankind for millennia and provided powerful tools to inform interventions. However, the foundational frameworks of computational causality, such as those based on Directed Acyclic Graphs, presuppose a fixed, linear, and Euclidean world. This monograph addresses the well-recognized challenge of extending causal reasoning to dynamic system engineering, where systems are often characterized by non-Euclidean data representations, non-linear temporal structures, and dynamic causality. Existing methods of computational causality, while valid and effective in their respective domains, were not designed for this class of complex, dynamic problem.

In response, this monograph presents the Effect Propagation Process (EPP): a theory of dynamic causality. The EPP adopts and formalizes a computable Causaloid, a concept first introduced by the physicist Lucien Hardy. Hardy proposed the Causaloid as a logical construct that merges the classical dualism of cause and effect into one unified entity that is agnostic of a fixed spacetime.

The EPP is foremost a theory of dynamic causality that became possible because of the adoption of the causaloid and its adoption lead to a cascade of novel capabilities: it allows the EPP to model causal relationships that are independent of any specific structure of space and time, to externalize context as a first-class entity, to introduce dynamics via contextual relativity, and to manage structural complexity through isomorphic recursive composition.

The EPP is therefore a synthesis that integrates the philosophical need for a relational foundation as suggested by Floridi, the statistical need for hierarchical models as advanced by Blei, and the physical need for spacetime-agnostic causality as pioneered by Hardy.

The EPP integrates insights and methods from three distinct domains. It begins with a philosophical foundation formulated in a dedicated metaphysics, ontology, and epistemology that establishes the first principles and dynamics of the EPP. Next, these principles are then translated into a rigorous formalization to ensure logical consistency and precision of the EPP. Finally, the theory is grounded in reality through its principled implementation in the open-source DeepCausality project hosted at the Linux Foundation.

This monograph is structured to guide the reader through the presented material. It begins by detailing the philosophical foundations before presenting the complete formalization. The subsequent chapters detail the implementation, including performance benchmarks and working examples in relativistic navigation. The monograph concludes by exploring the profound implications of this theory of dynamic causality and by arguing that the EPP's architecture provides a plausible path towards a computable "Effect Ethos" designed for verifiable alignment.

2 History of Causality

2.1 Plato

Plato is believed to be the first who explored the cause (aitia) in the book of Timaeus (c 360 BC). In it, Plato stipulates that multiple indispensable factors, the model, the maker (Demiurge), the material, and the space (receptacle), explain how the physical world with all the things in it are made[1]. Aristotle (c 350 BC) formalized the notion of causality in his Metaphysics[2] with the "Four Causes”[3]. These are:

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.

2.2 Seneca

Seneca (c 56 AD) argues in letter 65[4] 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 treating spacetime as a background for causal processes.

2.3 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[5] 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.

2.4 Albert Einstein

Albert Einstein (1879 - 1955) departed from Newtonian physics through his theory of General Relativity[6] (GR) in which he established that space and time are one manifold, spacetime, that is bent by gravitational influence from large masses. General Relativity preserves the prerequisite of a spatiotemporal context for causality, echoing Seneca’s key insight. However, the notion of dynamic spacetime requires a dynamic view of causality to fit into the dynamic spacetime manifold.

2.5 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 across space) and how that entire state evolves continuously and dynamically. Therefore, for Russell, the idea of classical causality, a strict happen-before relation, does not match the reality of modern physics anymore. Consequently, Russell wrote in his 1912 essay "On the Notion of Cause"[7]:

The law of causality, [...], is a relic of a bygone age" - Bertrand Russell

Many modern physics laws are time symmetric, and that means if state S1 at time t1 is related to state S2 at time t2 by a law, it’s equally true that state S2 at time t2 is related to state S1 at time t1. This relationship isn’t a simple, linear, one-way street from a necessary "cause" to a dependent "effect." Knowing the state at any time allows you 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 he observed in physics. His critique resonates with challenges encountered in contemporary quantum research.

2.6 Quantum Physics

Quantum Field Theory[8] (QFT) stands out as one of the most rigorously tested theories with unparalleled predictive accuracy in the history of science. QFT predictions in Quantum Electrodynamics have been experimentally verified up to an astonishing accuracy of one part in a billion or better. This experimental success proves that the underlying concepts such as the primacy of fields, the emergence of particles, and inherent indeterminacy used across different areas of QFT indeed result in demonstrably more accurate models.

The physics standard model, built on top of QFT, despite being one of the most successful theories of all time, accurately describes three of the four known fundamental forces: electromagnetism, the strong nuclear force, and the weak nuclear force. Notably absent remains gravity due to complex discrepancies between Einstein's (pre-quantum) theory of relativity and Quantum Field Theory. It is important to point out that General Relativity in itself is successful in the sense of high accuracy and strong predictive power, albeit in a non-quantum realm.

The unification of general relativity with quantum field theory would complete the standard physics model, but doing so faces a non-trivial impasse, as Lucian Hardy[9] formulated: "Quantum theory is a probabilistic theory with fixed causal structure. General relativity is a deterministic theory but where the causal structure is dynamic." Furthermore, the emergence of Quantum Gravity directly challenges the traditional separation of causality through the introduction of indefinite causal order with time symmetry[10].

Indefinite causal order means the order of causal events is not fixed. It might be that A happens before B, or that B happens before A, or a quantum superposition where both orders (A before B, and B before A) coexist simultaneously. Time symmetry or time reversal symmetry refers to the theoretical symmetry of physical laws under inverted time. That means, the laws of physics remain the same when time is reversed, a property known as T-symmetry. Many laws of physics are formulated in terms of time symmetry, thus allowing to calculate previous system states in time using the same set of equations. For example, laws like Newtonian mechanics, Maxwell's equations of electromagnetism, or Einstein's equations for General Relativity are T-symmetric.

The fundamental dynamical laws of quantum physics (excluding the weak nuclear force) are accepted as T-symmetric. However, the transition of a quantum state from a probabilistic T-symmetric state into a definite and time-asymmetric state, while observed, isn't well understood and subject to ongoing research in the field of quantum measurement.

Quantum superposition of states, on the other hand, is extremely well-observed, well-documented, and as a result, it is accepted as a fundamental building block of quantum physics. Quantum superposition inspires the exploration of concepts like indefinite causal structures in theories aiming to unify quantum mechanics and gravity. Therefore, the classical conceptualization of cause and effect embedded into a background spacetime might not yet exist at the fundamental level of quantum gravity. In Quantum Gravity, space and time are not external conditions, but potential emergent properties of the internal quantum structure itself. The problem is no longer whether spacetime is static or dynamic, but that spacetime itself emerges from the quantum level and thus positions itself as a higher-order effect of a generative process.

> Instead of asking "Where and When does this cause operate?", quantum gravity asks "What is the underlying process from which the notions of 'where' and 'when', and thus causal order, emerge?"

Russell saw physics moving towards laws governing states, a view echoed in quantum gravity's search for fundamental rules governing the structure from which spacetime and causal order emerge. At this stage, the understanding of causality evolved from a structure that required the existence of space towards a dynamic generative process from which causality emerges. This emergent causality does not rely on a pre-existing spacetime, but is grounded in a more fundamental level of reality—a set of underlying rules (i.e., conceptualized as a 'generating function') that determine the fundamental manifestation of spatiotemporal properties.

If the 'output' in terms of emergent properties doesn't include states that resemble classical spacetime, then the conditions Seneca deemed necessary for causality would not appear. From the quantum perspective, 'space and time' that Seneca identified as necessary conditions are not fundamental inputs to causality any longer, but are outputs from the underlying quantum generative process.

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's a shift from asking "What causes X given spacetime?" to "What process generates spacetime (and thus enables X to be caused)?".

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 [11].

A foundational framework for representing causal structures is based on graphical causal models, most notably Directed Acyclic Graphs (DAGs) [12, 13, 14, 15]. In these models, variables are typically represented by nodes, and directed edges indicate direct causal influences [13]. 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 [13, 16]. The theoretical underpinnings of Structural Causal Models (SCMs), which are closely related to graphical models, have been extensively studied [17, 18, 19, 20, 21]. Methods exist for handling complex scenarios, including incorporating latent variables [22, 23] and understanding the relationship between different causal models [24, 11]. Policy interventions in specific graphical structures, such as Lauritzen-Wermuth-Freydenburg (LWF) latent-variable chain graphs, have also been investigated [25]. This includes work providing a novel identification result for effects of policy interventions in these graphs [25].

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.

The dominant paradigm in modern computational causality is arguably the **Structural Causal Model (SCM)** framework, extensively developed by Judea Pearl and his colleagues [16]. 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 $X_j \to X_i$ 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 [16]. 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.

Alongside SCMs, the **Potential Outcomes Framework**, also known as the Rubin Causal Model (RCM) [26], offers another rigorous foundation for causal inference, with early conceptualizations by Neyman [27] and formally developed for observational studies by Rubin [28]. It has been particularly influential in statistics, econometrics, and 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 fundamental challenge, often termed the "fundamental problem of causal inference," is that only one of these potential outcomes can be observed for any given unit [26]. Causal inference in this framework often relies on assumptions such as the Stable Unit Treatment Value Assumption (SUTVA), which posits no interference between units and well-defined treatment versions, and ignorability (or unconfoundedness), which states that treatment assignment is independent of potential outcomes, conditional on observed covariates [29]. This framework excels in clarifying the conditions needed for estimating average treatment effects (ATE) from observational data, often employing methods like matching, stratification, or inverse probability weighting based on propensity scores [29].

While SCMs and the Potential Outcomes framework have different notational and conceptual starting points, they have been shown to be largely consistent and can often address the same set of causal questions, with SCMs providing a more explicit language for encoding causal mechanisms and deriving identifiability conditions through graphical criteria [16]. These foundational theories provide the bedrock upon which most modern computational causality techniques, including those involving deep learning, are built or against which they are compared.

3.2 Counterfactuals: Reasoning About What Might Have Been

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 [16]. 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 [16, 30].

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 [16]:

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 [16, 31]. Foundational work also explored the bounding and identification of specific types of counterfactual queries related to probabilities of causation [32]. Formal systems like Pearl's *do*-calculus provide tools for determining if causal effects under intervention are identifiable from observational data [33], and algorithms exist to automate this process [34], 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 [35, 36, 37, 38]. 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 [39], and investigates optimizing treatment effects in such settings [40]. Causal bandits also incorporate causal background knowledge into online decision-making problems [35, 36, 37, 38].

3.3 Causal Discovery

A central and challenging 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 [14, 41, 42]. 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 [43]. The KGS method [44], 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 [44]. 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 [45], 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 [46]. This work also deduced Karush-Kuhn-Tucker (KKT) optimality conditions for an equivalent reformulation of the NOTEARS problem [46]. Comparisons showed that a method called Abs-KKTS performed better than NOTEARS in terms of accuracy and computational efficiency on various graph types [46]. GraN-DAG is another approach that uses neural networks to handle non-linear causal relationships and utilizes a novel mechanism that uses interventional data to infer causal structures [47].

3.4 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 [48, 49]. 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 [50, 51]. 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) [52] 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 [53, 54]. 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?) [55].

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 [56]. 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.

Beyond these, 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.

3.5 The Role of Context and Temporality 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 [57] 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. [58, 59] 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 [48] and dynamic graphical models like DUCGs [52], 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 [56] 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.

3.6 Causal Inference and Discovery on Graph and Hypergraph Structures

The representation of causal relationships via graphical models, predominantly Directed Acyclic Graphs (DAGs) as foundational to Structural Causal Models (SCMs) [16], 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. [60] 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.

3.7 (Geometric) Deep Learning for Causal Inference and Representation

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 [61, 57]. 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 [62] 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.

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. [63] 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 signifies 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.

The comprehensive survey by Jiao et al. [57] also 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 in this domain is to move beyond enhancing specific sub-tasks towards building more integrated and principled frameworks for comprehensive causal reasoning using deep learning.

3.8 The Quantum Gravity Causaloid Framework

A significant effort towards establishing a framework for probabilistic theories with dynamic causal structure was presented by Hardy [64]. 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 [64]..

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 spacelike 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 [64]..

3.9 Deep Learning

Deep learning has achieved remarkable success in various tasks, including representation learning and prediction [65]. Related areas include neural architecture search [66, 67] and techniques for handling complex relationships in data, such as those explored using hypergraphs [68, 69]. Hypergraphs, introduced by Berge in 1973 [69], can model multi-way relationships and have found applications in areas like visualization [70, 71], partitioning [72, 73, 74], and recommender systems [75, 76, 77, 78, 79, 80]. Link prediction, particularly in multiplex networks, is another active area where deep learning is applied [81, 82].

More recently, researchers have explored the intersection of causal inference and deep learning, with the aim of leveraging the strengths of both fields [65]. 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 [14] [2]. Specific efforts include incorporating causal graphical prior knowledge into predictive modeling [83] and matching learned causal effects with domain priors in neural networks [84]. 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 [35, 36, 37, 38, 85, 86].

3.10 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.

3.10.1 DoWhy (Microsoft)

Developed by Microsoft Research, DoWhy[1] [87] 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 in data science projects.

3.10.2 EconML (Microsoft)

EconML[2] [88] 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. EconML is particularly powerful for applications in economics, personalized medicine, and targeted marketing where understanding individualized causal responses is critical.

3.10.3 CausalML (Uber)

Developed initially at Uber, CausalML[3] [89] 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

3.10.4 CausalNex (McKinsey)

CausalNex[4] 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.

[1] https://github.com/py-why/dowhy

[2] https://github.com/py-why/EconML

[3] https://github.com/uber/causalml

[4] https://github.com/mckinsey/causalnex

4 Motivation

In his seminal 1985 essay, "Programming as Theory Building," [90], Turing Award winner Peter Naur argued that the act of creating a significant software system is not one of mere coding, but of building a deep, explanatory theory. The source code is just a formal notation; the real program is the mental model of how it works and, more importantly, why it was built that way.

This monograph is a deliberate attempt to use programming and theory building as a joint force to apply the Naur principle synergetically where the implementation imposes new challenges on the theory and the theory guides a stronger, more principled implementation.

The pre-existing philosophy of causality served mankind for millennia, and one might be tempted to conclude that this is all there is to know about cause and effect. However, the origin of the Effect Propagation Process (EPP) did not start in philosophy, but in three distinct problems. The first problem is related to non-Euclidean data representation, the second problem is rooted in causal inference over non-linear time representations, and the third in handling dynamic causal structures.

4.1 Non-Euclidean representation

The first problem applies equally to computational causality and deep learning; therefore, it is best illustrated with familiar large language models. Language embeddings remain foundational to contemporary large language models (LLMs), but these require a reduction into a vector space because many prevalent LLM architectures operate efficiently in vector spaces, thus making the reduction from non-Euclidean realms (language) into a Euclidean representation (Vector space) mandatory. Instead of advancing LLM architectures to natively handle non-Euclidean representations, the industry has focused on leveraging Vector databases for storing and retrieving embeddings derived from LLMs. Graph neural networks operate on non-Euclidean spaces, but as these are not yet commonly adopted as core components in mainstream LLM architectures, the problem prevails.

4.2 Non-linear Time

However, when generalizing space beyond Euclidean, then the second problem emerges: how to represent time? More profoundly, can we separate time and space from data? Out of this insight, the idea emerged to separate space, time, and data into a separate context usable by multiple models. As space and time were elevated from an implicit background into an explicit first-class context representation, then moving beyond correlation towards causality became the next obvious step. At this moment, a profound problem emerged: When space is non-Euclidean, and time might not be a simple linear progression, then how do we establish a causal relationship?

As it turned out, establishing a clear causal relationship became problematic within the classical definition of causality, which fundamentally relies on a linear time-asymmetric ordering (cause preceding effect) within its assumed background spacetime.

One might challenge the presumption of non-linear time progression, but in complex systems with dynamic feedback loops, it's perfectly possible to see context structures that entail non-linear time regions. Non-linear time regions can occur when the background time is represented as a temporal hypergraph that holds multiple time resolutions simultaneously. The simultaneous presence of time units at different scales breaks the simple time-linear assumption (all time has the same unit, and moves therefore at the same rate) that computational causality tools commonly make.

Furthermore, in a temporal hypergraph, the unit of time is scale dependent which means in order to compare temporal values one must consider the scale to make a valid comparison between equally scaled values (hour X compared to hour Y). Less obvious, a temporal hypergraph, by design, holds all past and present temporal values simultaneously within its structure. This co-existence of multiple temporal points simplifies non-trivial temporal arithmetic over hetero-scaled time units, yet it imposes a vexing problem: How do you know if a time value in a node of the graph is current or past?

The problem is non-trivial because, as time progresses, the engulfing context engine continually generates the non-Euclidean temporal hypergraph representation with the implication that, at one lookup, the value of a temporal node is current, but at the next one, it might be past; however, the exact temporal distance at which a “present” value becomes “past” depends on the node’s time scale. A node holding a temporal value “hour” will be valid for 60 minutes; that means a lookup every ten minutes will yield the current hour 6 times, but handling the seventh lookup leads to a fundamental design decision that illustrates the implied complexity.

A “dynamic-position” design means, when a new hour starts, a new node will be added; therefore, the seventh lookup returns a past value. By implication, the index of the new node needs to be looked up to retrieve the value of the new current hour. Therefore, a “dynamic-position” design requires a dynamic index to locate current values.

A “static-position” design means, when a new hour starts, the previous current value will be overwritten with the understanding that the seventh lookup returns the new current value. By implication, the index of the current value always remains static. Therefore, a “static-position” design requires a fixed index, i.e., a lookup table to locate current values. Use cases exist for both scenarios and in practice, temporal hypergraphs use a mixture of static and dynamic positioning to handle different types of relative values, e.g., current day, last year, next hour, and similar..

Exacerbating the problem further, feedback loops across different time scales using different relative values may dynamically modify the temporal hypergraph itself to capture non-regular observations or to add estimations at future time values that have not yet occurred. At this point, it becomes abundantly clear that the assumption of a simple, unidirectional linear temporal order required for establishing cause-and-effect becomes untenable.

At the same time, causal relations remain valid in those non-linear regions. Additionally, the designation of a cause purely based on strict temporal order feels arbitrary in a temporal hypergraph in which past, present, and estimated future temporal values across multiple time scales all exist simultaneously. Regardless of static or dynamic location of relative values, the definition of causality had to evolve to match the reality of modeling causal structures across complex multimodal hypergraphs.

4.3 No a-priori causal structures

When modeling dynamic feedback loops across different time scales, the third problem emerges eventually: Not all causal structures are known upfront. There are cases where the causal structure emerges from the prevalent context predominantly in response to a change in externalities. For example, in the financial industry, a shifting volume imbalance indicates an emerging regime change. The exact cause for the shifting volume imbalance can be attributed to externalities such as breaking news. However, in absence of internal references, the subsequent causal structure emerges as part of the unfolding regime change. There is no possible way to know the new structure upfront therefore, it cannot reliably be modeled ahead of time. Likewise, in correlation-based methods, a similar phenomenon unfolds because, in absence of internal references, the deep learning model cannot predict correctly anymore because data during an emerging regime change fall outside its training data distribution. The mechanism is different, but the outcome is the same: that previously reliable models crater in novel situations.

The previously identified limitation of temporal order directly applies to dynamic regime changes because, if the new causal structure has not yet emerged, how can we know its temporal structure beforehand? In short, we cannot know reliably anything about emerging causality up to the moment it emerges.

These three problems, non-Euclidean representation, non-linear temporal structures, and emerging causality deeply interrelate with each other, thus defy overly simplistic solutions. For example, advanced graph neural networks work on non-Euclidean data representation, but fail on non-linear temporal structures. One might be tempted to build non-linear-time graph neural networks, but this does not address the problem of non-Euclidean data representation and emerging causality. There is research to combine methods from computational causality with deep learning, but these approaches are focused on non-Euclidean representation without integrating the challenges of non-linear temporality and emergent causal logic.

4.4 Boundaries of Classical Causal Models

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 purpose of the following analysis is to carefully delineate the set of assumptions, such as a fixed spacetime and a static causal structure, upon which they were designed to operate. By clearly defining these boundaries, we can identify the emerging class of problems in modern dynamic systems that now fall outside this classical scope.

Granger Causality

Granger Causality[91] is used for time-series data where past values of one time-series are used to predict values of another time-series. In a strict sense, Granger Causality tests if one variable (say X) predicts another variable (Y) through a series of t-tests and F-tests on lagged values of variable X.

Assumptions:

Granger causality assumes a stable causal structure and a linear, uniform time representation with consistent intervals.

Implications:

- **Non-Linear Time:** Granger causality cannot handle non-linear time representation
- **Non-Euclidean Representation:** Granger causality operates on time-series values within a Euclidean representation. It cannot be applied to a non-Euclidean representation.
- **Emergent Causality:** Because of the assumption of a stable causal structure, Granger causality cannot operate on emergent causal structures.

Pearl's Causal DAGs and Structural Causal Models (SCMs)

Judea Pearl pioneered the formalization of causality that underpins the majority of contemporary methods of computational causality. The framework of Structural Causal Models[17] (SCMs) rests upon the assumptions of Directed Acyclic Graphs (DAGs).

Assumptions:

Pearl's causal framework is exceptionally powerful for reasoning about interventions given a known or discovered causal model. The framework assumes:

- 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 embedded 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 DAG structure could potentially allow for non-Euclidean representation, but existing tooling assumes Euclidean structures and thus does not allow non-Euclidean representation
- **Emergent Causality:** The assumption of static causal structure prevents any handling of causal emergence.

Rubin causal model (RCM)

The Rubin causal model[92] (RCM) is a statistical framework for estimating causal effects by comparing potential outcomes under different treatment assignments.

Assumptions:

RCM assumes stable units, no interference between units (SUTVA), and that treatment assignment is ignorable conditional on observed covariates. RCM also assumes a stable causal structure and a fixed background spacetime.

Implications:

- **Non-Linear Time:** RCM is not designed to handle non-linear or multi-scale time representation.
- **Non-Euclidean Representation:** RCM is designed for variables representing treatments, outcomes, and covariates, without any mechanism for complex relational types. Therefore, RCM cannot infer across non-Euclidean data representations.
- **Emergent Causality:** RCM estimates effects within a stable causal structure and therefore cannot handle emergent causality.

Dynamic Bayesian Networks

Dynamic Bayesian Networks[93] (DBN) extend Bayesian Networks to model temporal processes by defining transition probabilities between states.

Assumptions:

Dynamic Bayesian Networks, similar to Granger causality, require a linear and uniform time representation. DBNs model changes in the state of variables over time according to a fixed probabilistic causal structure assumed to be static.

Implications:

- **Non-Linear Time:** DBN cannot handle non-linear time.
- **Non-Euclidean Representation:** DBNs reason over probabilities, not non-Euclidean space itself.
- **Emergent Causality:** DBNs assume a static causal structure and thus cannot handle emergent causality

As stated earlier, all these methods remain valid in their respective domains. Granger causality remains vital in time series forecasting. Pearl's causal DAG remains relevant, among others, in empirical healthcare. Dynamic Bayesian Networks find applications in healthcare, data mining, and robotics. This analysis, however, shows why each method cannot be applied to the previously established problems. It is important to recognize that, as Russell indicated, it is the classical interpretation of causality itself that warrants fundamental rethinking.

5 Causality as Effect Propagation Process

5.1 Background

The philosophy of the Effect Propagation Process (EPP) builds upon several important contributions in philosophy and science.

The EPP fundamentally rejects the classical Newtonian conception of a static absolute background spacetime. Historically, the idea finds precedent in Alfred North Whitehead who argued that reality is not composed of enduring substances but of dynamic, interconnected "actual occasions." This view inspired the shift towards a dynamic effect propagation process.

Luciano Floridi's view that the design principles for dynamic systems require a relational paradigm was profoundly inspirational to the formalization of the Effect Propagation Process. The EPP leverages the hypergraph as its foundational structure to model rich and complex relationships across all its elements. To do so, the ontology provides the foundational principles for handling disjoint categories of knowledge, not by exhaustively modeling all content, but by defining the foundational categories (e.g., space, time, data) and the dynamics of change itself.

Next, the EPP is inspired by Einstein's theory of General Relativity, which demonstrated that spacetime is a dynamic fabric, its geometry determined by the matter within it, which in turn dictates 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 metaphysical analogue of this profound physical insight.

The EPP's Emergence and Dynamic Co-Emergence principles are inspired by the profound work of Lucien Hardy on formulating a framework for Quantum Gravity. Hardy noted the core conflict: general relativity is a theory with a dynamic causal structure, while quantum theory operates on a fixed causal structure. More profoundly, Hardy points out that Quantum Theory leans probabilist causal structures whereas General Relativity uses deterministic causal structures. This insight inspired the multi modal reasoning across probabilistic and deterministic representations of causality in the EPP.

Hardy introduced the "causaloid," a concept that encapsulates a region of spacetime and the causal connections within it. The EPP draws direct inspiration from Hardy's pioneering work by using the term Causaloid honoring Hardy's concept of a unified, self-contained unit of causality, though it has been adapted for a more general, computational context. Furthermore, the EPP's concept of Dynamic Co-Emergence is a direct architectural answer to the challenge of indefinite causal structures Hardy identified. The EPP's Emergence principle provides a computational mechanism to dynamically generate and select a new causal structure (a new archê kai aitia) that did not exist before in response to a trigger event. This co-evolution of the causal structure and its context is a computational mechanism for realizing a system with dynamic indefinite causal structures that, on demand, become into being.

The contribution of the EPP ontology is to synthesize all these best in class concepts and formalize them into a deeply integrated ontology that defines the specific primitives (Causaloid, Context), compositions (Isomorphic Recursion) before introducing Contextual Relativity as the governing principle of predictable dynamics and the Generative Process as the governing principle of dynamic emergence.

5.2 Definitions

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 classical definition of causality, taken from Judea Pearl's foundational work[17]:

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

When removing time 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 arbitrary, 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. 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 deals with 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 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 describe a cause in terms of its effects. Therefore, it is true that causality deals with effects.

The third one, effect propagation, stretches the imagination and is less obvious. When we rewrite the previous definition of classical causality in terms of effect propagation, 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. Therefore, causality becomes an effect propagation process.

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. 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 you designate the preceding effect to be a “cause”, then you can rewrite the general definition back into the classical definition thus the general and the specialized definition of causality remain congruent. This detail is important because the generalized effect propagation process definition would not be sound if it were unable to express a specialized variation. Framing causality as an effect propagation process leads to several implications:

5.3 Effect Propagation

In EPP,’effect propagation’ denotes the operation of a specific, definable mechanism (via the causaloid) that links states through the underlying fabric regardless of how that fabric might be defined.

The first part, the causaloid, operationalizes effect transfer within a system without relying on any metaphysical causal power. The effect transfer encapsulated in the causaloid is intrinsic to its mechanistic definition. The identification of relevant causaloids requires explicit modeling and hypothesis testing or (future) discovery processes. EPP provides the formatl structure for representation of Causeloids once discovered.

The second part, the non-defined fabric, through which effects propagate, serves the purpose of externalizing the fabric as a specific, definable context. Instead of assuming an implicit background spacetime, the EPP externalizes the fabric through which effects propagate as a specific context that is defined by a different generative function. Therefore, EPP fundamentally moves away from invoking a metaphysical causal power towards definable, testable, and verifiable functions (Causaloids) that define the functional relationships describing how a system changes if these functions are active.

5.4 Spacetime Agnosticism

In quantum gravity, where spacetime geometry might be in a superposition or non-existent at the fundamental level, "propagation" isn’t a movement along a geodesic in a manifold. It’s the propagation of effects through a network of states or elements defined by the fundamental structure. This is a prerequisite for handling indefinite causal order.

5.5 Indefinite Causal Order

In situations where the causal structure itself is indefinite (a superposition on the quantum level, a not-yet-emerged state in GR), "effect propagation" can be understood as the influence propagating through a superposition of possible pathways through the fundamental structure. The "effect" isn't tied to a single, definite causal link but is a result of the propagation through all possible pathways.

5.6 Causaloid

In the Effect Propagation Process framework, due to the detachment from a fixed spacetime, this 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 one entity. Instead of dealing with two nearly identical concepts discernible from each other by temporal order, the causaloid is one concept that defines causality in terms of its effect transfer without presupposing a fixed spacetime background. 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 uses Hardy's formal definition. Instead, the EPP formulates the Causaloid as an abstract structure, thereby decoupling it from any particular physical theory while preserving its core philosophical utility and making it practical implementable in software.

In the post-quantum context, the term "propagation" does not imply movement through a pre-existing space or time in the classical sense. Instead, it refers to the fundamental process by which an effect is transferred within the underlying structure of reality itself. 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 is 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.

Consequently, causality is understood as an effect propagation process that emerges from the fundamental structure or set of rules (akin to a generating function) from which spatiotemporal relationships emerge. This philosophical concept of the Effect Propagation Process finds support in physical theories that propose fundamental structures underlying spacetime, such as Causal Set Theory, and generalizes the idea of effect transfer present in Quantum Field Theory.

Lastly, the Effect Propagation Process offers a philosophical interpretation for mathematical tools that describe non-classical causal behavior, such as Process Matrices. However, the Effect Propagation Process remains agnostic to the operational details of any particular physics theory and, instead, it offers a coherent way of thinking about causality that aligns with the absence of finite spacetime in the quantum realm.

6 The Metaphysics of the Effect Propagation Process

The metaphysics of the Effect Propagation Process establishes a set of principles underlying the ontology. We begin by defining the three core metaphysical concepts: Monoidic Primitives, Isomorphic Recursive Composition, and Contextual Relativity and then proceed to derive from these first principles the metaphysic of dynamics and becoming of the EPP. Conbined, these form the foundation of ontological 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.

6.1 Monoidic Primitives

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

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 is fundamental to the composability of the EPP.

6.2 Isomorphic Recursive Composition

Isomorphism means "having the same shape" in the sense that isotropic 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-reducible of form of monoidic primitives within the isomorphic recursive composition, it must have a singleton representation of itself.

6.3 The Metaphysics of Being

The metaphysics of being is structures in a classical Aristotelian notion (Book Zeta[94]):

- 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 amount form an Aristotelian hylomorphic compound. The 'Monoidic Primitives' constitute the "matter" (*hyle*) of 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.

6.4 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 defines 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.

6.5 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 mater, a new form, both of it, 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 p̀osits 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 becomes 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 refers to 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 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.

6.6 Discussion

The metaphysics of the Effect Propagation Process defines the substance of the EPP, its dynamics, and its emergence. Metaphysically, the principal 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 principal 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 crisis:

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 chose one state of emergence over another one 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 crisis 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 its 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 if its genesis process or not? The author argues for a unequivocal no. Considering the alternative, when a system that can evolve its own genesis process, it fundamentally becomes 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 possible have the innate ethos of a human being and thus cannot possible 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 conflict demands from different stakeholders? These are immense normative and political challenge and, a metaphysics alone, cannot possibly answer a fundamentally societal set of questions.

As a consequence, the genesis process itself is a decision with far fetching higher order effects. The mitigation of 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.

6.7 Summary

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

1. Substance (Being): The hylomorphic compound of Monoidic Primitives and Isomorphic Recursive Composition.
2. Dynamics (Changing): Governed by the principle of Contextual Relativity.
3. Emergence (Becoming): Governed by the principle of Higher-Order Emergence

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 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. The EPP, through the introduction of Higher-Order Emergence, establishes a foundation for exploring the interrelation between emergent systems and human value. However, it also presents a new set of unique challenges that demand a new epistemology and ontology of the EPP.

7 The Ontology of the Effect Propagation Process

7.1 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.

7.1.1 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 isa 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 Isomorphic 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 ccausaloid 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 isomorphic recursive composition means that any number of causaloids of different types uniformly compose. A complex causal graph can be encapsulate 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 build and tested in isolation, or as part of a specific sub-graph. 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 mew singleton causaloid. Likewise, an arbitrary 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.

7.1.2 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, EuclideanSpace alongside another representing a point in a non-Euclidean GeoSpace. From the perspective of the ontology, both are valid "spatial fact 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 contextuid of its structure.

7.1.3 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 isomophic 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 lifetimes. 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 build 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 maintain 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% higher?"), and then evaluating the same Causaloid logic against these altered frames of reference. This can be achieved through architectural patterns such as "hot/cold" context partitioning, where the subset of Contextoid relevant for the counter factual analysis are centralized into one dedicated contest, from which the alternate versions are derived. When combined with high-performance data structures, this enables the execution of thousands of counterfactual simulations in real-time, even on resource-constrained embedded devices.

7.1.4 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, were 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 isomorphic 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 enables 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.

7.1.5 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.

7.2 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 properties arising relative to its context with the implication that the same object may have different states when used in different context, but, more profoundly, the same object may alters its state when context itself is relativistic altered and populates its contextual adjustment to all connected monoidic primitives. Contextual relativity represents the duality a monoidic primitive being influenced relative to its context while simultaneously the context itself is being influenced by relativistic effects of the engulfing fabric itself.

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 on 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 defined 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 applies it to the existing value.

Update means and existing value is replaced with a new value, fort example a new sensor reading. The core property of an update operation is that to take a as 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. 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's getting closer to the thermal limit. Critically, when transferring the model to a different opertatonal environment that requires a different thermal threshold, the replacement of one contextoid is sufficient to ensure the correct functionality of the modeled thermal system.

7.3 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 of a principled implementation at a later stage.

7.3.1 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 becmomes necessary. Triggers can originate from external stimuli, the passage of time, or an explicit external commands. 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.

7.3.2 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 factt 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 is not an atomic command. Instead, Evolve is a meta-command, it's 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 the 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.

7.3.3 The Generative Process

The Generative Process is the operation responsible for actuating the Generative Command. It interprets xommand 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.

7.3.4 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 attest, the 'what happened" during the process in 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.

7.4 Discussion

The ontology of 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

8 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.

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.

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 EPP 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.

Knowledge Derivation

The EPP derivates 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.

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 pre-requirements to support the full explainability of each inference. Furthermore, in the case of multi-stage reasoning, the sequence of applied Causaloids establishes the order or 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 is rooted 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.

8.1 Static EPP Epistemology

For a static Effect Propagation Process, the knowledge is explicitly modeled during the design stage and confined in 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’s 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. 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.

8.2 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 of the epistemology remains deepening not only the EPP evolution itself but 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 determines fundamentally what 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 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). One could estimate a potential epistemology by using a Rubin causal model[92] (RCM) by comparing potential reasoning outcomes under different scenarios in which a Causaloid was generated versus when not.

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 dynamic evolving behavior.

8.3 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 Causaloids 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 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 EPP's core design philosophy.

Regardless of the mechanism, 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 determines fundamentally what 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 emerging 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 representation 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.

8.4 Operationalized meaning of truth in EPP

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. That 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 amidst 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.

8.5 Causal Emergence

The problem of modeling 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 discovery

Instead of trying to find a fixed causal structure, EPP models how causal relationships emerge from underlying (dynamic) processes. The existing work on causal discovery remains valid; the EPP, however, takes the idea takes the idea one step further by incorporating a dynamic generative process. Further research will verify the utility of this perspective, but at least it expands the notion of discovering a static structure to describing a dynamic process.

Causal transferability

Instead of trying to capture the exact conditions under which a causal relationship holds true, the EPP specifies all presumptions as a generative function, which makes it fundamentally testable and thus transferability can be decided.

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 modality. 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 from 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, EPP itself has to be spacetime-agnostic.
- **Hardy's Causaloids are necessary:** Because the EPP itself is spacetime-agnostic, a different representation of causal relationship 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.

The Effect Propagation Process framework constitutes a foundational shift to viewing causality itself as emergent and thus redefines what causality means in dynamic systems. Because of its flexibility, EPP can express static causal relationships similar to Pearl's Causal DAG, it can handle dynamic causal systems similar to Dynamic Bayesian Networks, but then goes further and enables dynamic causal emergence. Dynamic causal emergence has real-world applications:

- **Financial Markets:** Causal relationships between assets change based on market conditions. EPP can be used to model how these relationships emerge and dissolve.
- **Biological Systems:** Gene regulatory networks where modulating relationships emerge based on cellular state.
- **Social Systems:** How influence relationships emerge and change in social networks.

9 The Formalization of the Effect Propagation Process

9.1 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.

9.1.1 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, or a general data value.

Contextoid Definition

A Contextoid v is an element of a set of all possible contextoids $V_{\mathcal{C}}$ within a given Context Hypergraph (defined in Section 9.1.2). It is formally defined as a tuple:

$$v = (id_v, \text{payload}_v, \text{adj}_v)$$

Contextoid Identifier

Let $\mathbb{I}$ be a universal set of unique identifiers. Then $id_v \in \mathbb{I}$ is a unique identifier for the contextoid v. This ensures each piece of contextual information can be uniquely referenced within its encompassing Context Hypergraph C.

Contextoid Payload

The payload_v represents the actual contextual information encapsulated by the Contextoid. It is defined as a tagged union, allowing for diverse types of context:

$$\text{payload}_v \in \{\text{Data}(d) \mid d \in \mathcal{D}_T\} \cup \{\text{Time}(t) \mid t \in \mathcal{T}\} \cup \{\text{Space}(s) \mid s \in \mathcal{S}\} \cup \{\text{SpaceTime}(st) \mid st \in \mathcal{ST}\}$$

where $\mathcal{D}_T, \mathcal{T}, \mathcal{S},$ and $\mathcal{ST}$ are predefined sets representing the domains of all possible data, temporal, spatial, and spatiotemporal values, respectively.

Data Payload ($Data(d), d \in \mathcal{D}_T$) The set $\mathcal{D}_T$ represents the domain of all possible data values that a Data Contextoid can hold. The specific nature of d is generic (of type T) and can encapsulate arbitrary information relevant to the causal model, such as sensor readings, calculated metrics, textual features, or symbolic states.

Time Payload ($Time(t), t \in \mathcal{T}$) The set $\mathcal{T}$ represents the domain of all possible temporal values. A temporal value $t \in \mathcal{T}$ is typically structured as an ordered pair:

$$t = (\text{scale}, \text{unit})$$

Formalizing Temporal Scale (T_{scale}) and Unit (T_{unit}) Let $\mathcal{T}_{\text{scale}}$ be an enumerated set of possible temporal granularities, e.g.,

$$\mathcal{T}_{\text{scale}} = \{\text{Year, Month, Day, Hour, Minute, Second, Nanosecond, EventStep}\}$$

Let $\mathcal{T}_{\text{unit}}$ be the domain of values for the temporal unit, typically non-negative integers ($\mathbb{N}_0$) or a suitable ordered set, representing the count at the given scale. Thus, scale $\in \mathcal{T}_{\text{scale}}$ and unit $\in \mathcal{T}_{\text{unit}}$.

Space Payload ($Space(s), s \in \mathcal{S}$) The set $\mathcal{S}$ represents the domain of all possible spatial values. A spatial value $s \in \mathcal{S}$ can be represented by coordinates, for example, as a tuple for up to three dimensions:

$$s = (x, y, z)$$

where x, y, z are optional components.

Formalizing Coordinates (T_{coord}) Let T_{coord} be the domain for coordinate values (e.g., $\mathbb{R}$, $\mathbb{Z}$, or a generic type V). Each coordinate component $x, y, z \in T_{coord} \cup \{\text{null}\}$, allowing for variable dimensionality.

Formalizing Spatial Interpretation (Euclidean/Non-Euclidean via `Spatial<V>`) The geometric interpretation of spatial coordinates (e.g., distance metrics, neighborhood relations) is not fixed but is governed by functions associated with the specific instantiation of the spatial context. In a computational framework like DeepCausality, this is typically managed through a trait or interface, analogous to the `Spatial<V>` trait discussed in the EPP philosophy [95]. This allows s to represent points in Euclidean spaces, nodes in a graph (non-Euclidean), or other abstract relational structures.

SpaceTime Payload ($SpaceTime(st), st \in \mathcal{ST}$) The set $\mathcal{ST}$ represents the domain of all possible spatiotemporal values. A value $st \in \mathcal{ST}$ combines spatial and temporal information, typically as a composite structure:

$$st = (\text{space\_value}, \text{time\_value})$$

where space_value $\in \mathcal{S}$ and time_value $\in \mathcal{T}$. This represents a specific spatial configuration at a particular temporal point or interval.

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.

9.1.2 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.

Context Hypergraph Definition. An individual Context Hypergraph C is defined as a tuple:

$$C = (V_C, E_C, ID_C, \text{Name}_C)$$

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.

9.1.3 Context Collection (C_{sys})

In many complex scenarios, causal reasoning may need to draw upon multiple, potentially distinct but interacting, contextual realms. EPP accommodates this through the concept of a **Context Collection**.

Definition 3.4 (Context Collection): A System Context, or Context Collection, C_{sys}, is defined as a finite set of distinct Context Hypergraphs:

$$C_{sys} = \{C_1, C_2, \ldots, C_k\}$$

where each C_i is an individual Context Hypergraph as defined in Definition 3.2. Each $C_i \in C_{sys}$ possesses a unique identifier $ID_{C_i} \in \mathbb{N}$ (distinct from other Context Hypergraphs in the collection) and a descriptive Name_{C_i}.

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

To enable Causaloids (defined in Section 9.2) to interact with the Contextual Fabric, a mechanism for querying and retrieving contextual information is required. This is conceptualized as a **Context Accessor**.

Definition 3.5 (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$(id_v, ID_C) \rightarrow v \cup \{\text{null}\}$: Retrieves a Contextoid v by its identifier id_v from a specific Context Hypergraph C (identified by ID_C) within $\mathcal{C}_{refs}$.
- getHyperedges$(v_i, \text{kind}_e) \rightarrow \{e_1, \ldots, e_m\}$: 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.

9.2 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, and these are organized into CausaloidGraphs (G) to model intricate webs of causal relationships.

In the EPP philosophy, Causaloids are grounded in a process of observation, assumption validation, and inference. While a full formalization of this empirical grounding process is extensive, we briefly define its key components as they inform the structure of a Causaloid, particularly its linkage to underlying evidence or hypotheses. Let $\mathbb{I}$ be a universal set of identifiers.

Observation Instance (o_data)

An **Observation Instance** o_{data} represents an empirical data point or a collection of related measurements relevant to a potential causal link. It typically includes an identifier, the observed value(s), and any associated outcome or effect. Formally, one might define $o_{data} = (id_{obs} \in \mathbb{I}, \text{val}_{obs}, \text{eff}_{obs})$. A dataset $\mathcal{D}_{obs}$ would be a set of such observations.

Assumption Instance (A_smp)

An **Assumption Instance** A_{smp} articulates a condition believed to hold, under which causal interpretations are made. It includes an identifier, a description, and critically, an evaluation function f_{asmp} that tests the assumption against observational data $\mathcal{D}_{obs}$ and/or contextual information $\mathcal{C}_{sys}$. Formally, $A_{smp} = (id_{asmp} \in \mathbb{I}, \text{desc}_{asmp}, f_{asmp}, \text{status}_{asmp})$, where $f_{asmp} : \mathcal{P}(\mathcal{D}_{obs}) \times \text{ContextAccessor}(\mathcal{C}_{relevant}) \to \{\text{true, false}\}$. ($\mathcal{P}(\mathcal{D}_{obs})$ denotes the power set or a relevant subset of observations).

Inference Instance (I_inf)

An **Inference Instance** I_{inf} represents a tested hypothesis about a potential causal link, derived from observations under validated assumptions. It typically includes an identifier, the inferential question, the strength of the observed relationship, and a threshold for asserting the inference. Formally, $I_{inf} = (id_{inf} \in \mathbb{I}, \text{question}_{inf}, \text{strength}_{obs}, \text{threshold}_{inf}, \text{status}_{inf})$.

9.2.1 The Causaloid (χ)

The fundamental unit of causal interaction within EPP is the **Causaloid**, inspired by Hardy's work [9] but adapted here as an abstract, operational entity for generalized causal modeling. It encapsulates a single, testable causal mechanism or hypothesis, whose activation depends on input observations and contextual information. Its definition reflects the versatile structural forms found in the DeepCausality implementation.

Causaloid Definition

Definition 4.1 (Causaloid): A Causaloid χ is defined as a tuple:

$$\chi = (id_\chi, \text{type}_\chi, f_\chi, \mathcal{C}_{refs}, \text{desc}_\chi, \mathcal{A}_{linked}, I_{linked})$$

where:

- $id_\chi \in \mathbb{I}$ is a unique identifier for the Causaloid.
- $\text{type}_\chi \in \{\text{Singleton, Collection, Graph}\}$ specifies the structural nature of the Causaloid, determining how its causal function f_χ is realized. This corresponds to the `CausalType` enum in the Rust implementation.
- f_χ is the **causal function or logic** associated with the Causaloid. Its precise signature and operation depend on type_χ (detailed in Section **??**). It fundamentally maps inputs and context to an activation status $\{\text{true (active)}, \text{false (inactive)}\}$.
- $\mathcal{C}_{refs} \subseteq \mathcal{C}_{sys}$ is a set of references to Context Hypergraphs that provide contextual information for the evaluation of f_χ.
- desc_χ is a human-readable description (e.g., a string) of the causal mechanism or hypothesis. This corresponds to the output of an `explain()` method in an implementation.
- $\mathcal{A}_{linked}$ is an optional set of identifiers $\{id_{asmp_1}, \dots\}$ referring to Assumption Instances (Section **??**) upon which this Causaloid's validity depends.
- I_{linked} is an optional identifier id_{inf} referring to an Inference Instance (Section **??**) that may have led to this Causaloid's formulation.

Causal Function Logic (f_chi)

The causal function f_χ embodies the specific operational logic of the Causaloid χ. Its behavior is contingent on type_χ:

- If type_χ = Singleton: f_χ is a direct evaluation function, $f_{\chi,S} : \mathcal{O}_{\text{type}} \times \text{ContextAccessor}(\mathcal{C}_{refs}) \to \{\text{true}, \text{false}\}$. This function directly tests a specific causal hypothesis against an input observation (of type $\mathcal{O}_{\text{type}}$, often a `NumericalValue` in implementations) and the accessed context.
- If type_χ = Collection: The Causaloid χ encapsulates an ordered or unordered set of other Causaloids, $\mathcal{X}_{coll} = \{\chi'_1, \chi'_2, \ldots, \chi'_p\}$. In this case, f_χ represents an aggregate reasoning logic over the activation states of the Causaloids in $\mathcal{X}_{coll}$. For example, $f_{\chi,C}$ might evaluate to true if all $\chi'_j \in \mathcal{X}_{coll}$ are active, or if any one of them is active. The evaluation of each χ'_j would itself involve its own causal function $f_{\chi'_j}$. The input $\mathcal{O}_{\text{type}}$ might be a collection of observations, one for each member of $\mathcal{X}_{coll}$, or a shared observation.
- If type_χ = Graph: The Causaloid χ encapsulates an entire CausaloidGraph $G' = (V_{G'}, E_{G'}, \ldots)$ (as defined in Section 9.2.2). Here, f_χ represents the outcome of the full Effect Propagation Process Π_{EPP} (defined in Section **??**) operating on the encapsulated graph G'. For instance, $f_{\chi,G}$ might evaluate to true if a specific target node in G' becomes active after propagation, or if the overall graph reaches a certain state. The input $\mathcal{O}_{\text{type}}$ would serve as the initial trigger(s) for propagation within G'.

Regardless of type_χ, the causal function f_χ ultimately determines the binary activation state of χ, providing the "testable effect transfer" mechanism central to EPP.

9.2.2 The CausaloidGraph (G)

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

CausaloidGraph Definition

Definition 4.2 (CausaloidGraph): A CausaloidGraph G is defined as a tuple:

$$G = (V_G, E_G, ID_G, \text{Name}_G)$$

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).

Set of Causal Nodes (V_G)

Each causal node $v_g \in V_G$ is defined as a tuple $v_g = (id_g, \text{payload}_g)$, 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.

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:

$$e_g = (V_{\text{source}}, V_{\text{target}}, \text{logic}_e)$$

where:

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

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:

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

This state evolves as effects propagate through the graph according to the Causaloid functions and the logic of the hyperedges, as will be detailed in Section **??**.

9.3 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.

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 $S_G(v_g) = \text{active}$ 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_χ, and its *consequence* is how its activation (or inactivation) influences other parts of the CausaloidGraph.

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 $\mathcal{O}_{\text{type}}$ 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.

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, $S_G : V_G \rightarrow \{\text{active, inactive}\}$ is its current state, $\mathcal{C}_{sys}$ is the Context Collection, O_{trig} is the set of current input triggers, and $S'_G : V_G \rightarrow \{\text{active, inactive}\}$ 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($\mathcal{C}_{refs_{\chi_{v_g}}}$).

   (b) The influence of source nodes V_{source} on target nodes V_{target} is mediated by the logic_e of the connecting hyperedge $e_g = (V_{\text{source}}, V_{\text{target}}, \text{logic}_e)$. 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.

9.3.1 The Operational 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 5.3 (Context Generative Function):

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

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 [95]).

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 5.4 (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)]

In the most general case, context and causal structure can co-evolve.

Definition 5.5 (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.

9.3.2 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 3.1), Context Hypergraph (Def 3.2), Causaloid (Def 4.1), nor CausaloidGraph (Def 4.2) 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, SpaceTime) are used within Contextoids. The core propagation function Π_{EPP} (Def 5.2) and generative function Φ_{gen} (Defs 5.3-5.5) 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 $S_G \to S'_G \to S''_G \dots$ 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.

[Handling Indefinite Causal Order

While a full formal treatment of quantum indefinite causal order is beyond the scope of this foundational paper, the EPP framework's spacetime agnosticism and its use of Causaloids (which do not inherently encode a fixed temporal role as "cause" or "effect") provide conceptual compatibility. Future work could explore:

- Extending the CausaloidGraph G to allow for superpositions of different hyperedge structures (E_G) representing different causal orderings.
- Modifying the propagation process Π_{EPP} to evaluate such superposed pathways, perhaps drawing from formalisms like process matrices.

The EPP's detachment from a fixed causal structure is a prerequisite for such extensions.

9.4 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. This section formalizes the CSM, reflecting its implementation within systems like DeepCausality[5].

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

Definition 6.1 (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:

$$q = (id_q, \text{data}_q, \chi_q, \text{version}_q)$$

where:

- $id_q \in \mathbb{I}_{CSM}$ is a unique identifier for this Causal State.
- $\text{data}_q \in \mathcal{O}_{\text{type}}$ is the specific input observation or data value (compatible with the Causaloid's expected observation type $\mathcal{O}_{\text{type}}$) 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 9.2.1), $\chi_q = (id_{\chi_q}, f_{\chi_q}, \mathcal{C}_{refs_q}, \dots)$. The function f_{χ_q} embodies the predicate for this state.
- $\text{version}_q \in \mathbb{N}$ is an optional version number for the state definition.

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

[5] deepcausality.com

Definition 6.2 (Causal Action): A **Causal Action** a represents a deterministic operation. It is defined as a tuple:

$$a = (\text{exec}_a, \text{descr}_a, \text{version}_a)$$

where:

- exec_a : void → 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.
- $\text{version}_a \in \mathbb{N}$ is an optional version number for the action definition.

The successful execution of exec_a yields Ok(void); failure yields an Err(ActionError).

Definition 6.3 (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 $id_q \to (q, a)$.

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 $\mathcal{SA}_M$.

9.4.1 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 $\text{trigger}_j = f_{\chi_{q_j}}(\text{data}_{q_j}, \text{ContextAccessor}(\mathcal{C}_{refs_{q_j}}))$.
   (b) If 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.

9.4.2 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): $\mathcal{SA}'_M = \mathcal{SA}_M \setminus \{(q, a) \mid id_q$ is the identifier of $q\}$.
- **UpdateStateAction**$(M, id_q, q_{updated}, a_{updated})$: Replaces the pair (q, a) associated with id_q with $(q_{updated}, a_{updated})$.

9.5 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 [17], here adapted to EPP principles.

9.5.1 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.

9.5.2 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)
  - $\text{type}_{\chi_{S \to T}} = \text{Singleton}$
  - $f_{\chi_{S \to T}} : \mathcal{O}_{\text{nicotine}} \to \{\text{true, false}\}$, 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):**
  - $id_{\chi_{T \to C}} = 2$
  - $\text{type}_{\chi_{T \to C}} = \text{Singleton}$
  - $f_{\chi_{T \to C}} : \mathcal{O}_{\text{tar}} \to \{\text{true, false}\}$, 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 $\geq$ 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.

9.5.3 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 $S'_G(v_{g1}) = \text{active} \wedge S'_G(v_{g2}) = \text{active}$. 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.

9.5.4 Formalizing Observations

This simplified example illustrates:

- **Causaloids** (χ) as operational units (Section 9.2.1), encapsulating specific causal functions (f_χ).
- A simple **CausaloidGraph** (G) structure (Section 9.2.2) 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.

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 hat 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 design 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 sub-graphs 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 lead to the decision to implement a hypergrah in a dedicated Rust crate called UltraGraph. Initially, UltraGraph wrapped the MatrixGraph implementation of the Petgraph[6] 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 arises. And indeed, as the requirements of the DeepCausality project keep 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 the NWHypergraph (NWHy)[96] 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 lifecycle.
- 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 recognize 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's 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 on 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 ca 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 eliminating cache misses and unlocking near-linear scaling. If the graph needs to evolve further, simply cal *.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 carefully 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

```rust
#[derive(Default)]
pub(crate) struct CsrAdjacency<W> {
    pub(crate) offsets: Vec<usize>,
    pub(crate) targets: Vec<usize>,
    pub(crate) weights: Vec<W>,
}
```

[6] 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 of 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 one 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

```rust
#[derive(Clone)]
pub struct CsmGraph<N, W>
where
    N: Clone,
    W: Clone + Default,
{
    // Node payloads, indexed directly by ‘usize‘.
    nodes: Vec<N>,
    // CsrAdjacency structure for forward traversal (successors).
    forward_edges: CsrAdjacency<W>,
    // CSR structure for backward traversal (predecessors).
    backward_edges: CsrAdjacency<W>,
    // Index of the designated root node.
    root_index: Option<usize>,
}
```

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 lifecycle. 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 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 1: 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[7].

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 2: 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[8].

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 massive 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). Developing such a cache-friendly concurrency model is subject to future work.

[7] `https://github.com/deepcausality-rs/deep_causality/tree/main/ultragraph/benches`

[8] `https://github.com/deepcausality-rs/deep_causality/tree/main/deep_causality/benches`

10.2.5 Discussion

This new implementation of UltraGraph based on its innovative two stage graph life-cycle and it 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 node. At this scale, the underlying CSR implementation will most likely benefit from contemporary reram-based[97] hardware accelerators specifically designed to accelerate hypergraphs with sparse representation. Furthermore, the two state 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

11 Validity

The validity of the Effect Propagation Process must be assessed according to the nature of its contribution. The EPP is a philosophy-informed, formal computational framework designed to provide a new, more expressive language for modeling dynamic, contextual causal systems. Therefore, its validity rests on the criteria appropriate for such a framework, analogous to how one would assess a mathematical logic or a new programming language:

- **Internal Validity** (Soundness & Consistency): The framework's internal validity is determined by its logical and mathematical soundness and coherence. As detailed in the accompanying specification, the EPP is built on a foundation of first-principles reasoning to ensure its components are consistent and its operations are sound.
- **External Validity** (Expressive Power & Scoped Generalization): The framework's external validity is demonstrated by its robustness against alternative interpretations of its premises and how well it defines the boundaries of its generalization.

11.1 Internal validity

The **soundness** of EPP derives from the first principled logical progression of the presented argument. The stated problem of inadequacy of classical causality in light of the challenges imposed by Quantum Gravity (emergent spacetime, indefinite causal order) is well-recognized and documented[10].. From this recognized problem, the argument for EPP progresses as stated below:

1. Establishes classical causality and its historical critiques.
2. Introduces the challenges from modern physics (GR, QG).
3. Shows why these challenges render classical causality insufficient.
4. Proposes EPP as a coherent response to these challenges.
5. Contrasts EPP with classical causality.
6. Discusses its ontological, epistemological, and teleological implications.
7. Acknowledges and addresses threats to its validity.

The soundness is further strengthened by its inspiration from established scientific theories (QFT, GR) and foundational work in quantum gravity. While quantum gravity as a scientific theory remains a work in progress, the conceptual challenges that arise from it are valid regardless of how any particular theory may explain the underlying quantum mechanisms.

The **consistency** of the EPP framework arises from a handful of carefully stated conclusions.

11.1.1 Spacetime Agnosticism & Causaloids

Premise: On a quantum level, spacetime may not exist.

Conclusion: Therefore, remove spacetime from EPP.

Premise: EPP does not have a defined spacetime.

Conclusion: Cause and effect cannot be separated anymore because there is no a priori temporal order.

Premise: Cause and effect cannot be discerned because of missing temporal order.

Conclusion: Fold cause and effect into one entity, the causaloid, that is independent of temporal (and spatial) order.

Note, the last conclusion holds because of the temporal order required for classical causality. The only logical alternative conclusion from the premise of missing temporal order would be to abandon causality altogether. However, this conflicts with the reality in which causal relationships indeed exist; therefore, the alternative has been deemed unsound.

11.1.2 Effect Propagation as the Essence of Causality

Premise: The causaloid, as a building block, is independent of temporal (and spatial) order.

Conclusion: Define causality by what it does (propagate effects) instead of what it was thought to be (a temporal order dependent atomic relationship).

The careful reader may raise a concern over the choice of words (i.e., propagate effects), since "transferring information" or "transmitting influence" might be equally valid choices. True, that is indeed a fair point. However, the term information has specific meaning in information theory and computer science, and likewise, the term influence has specific meaning in social science; therefore the author settled cautiously on "effect" mainly to prevent conflating different meanings.

11.1.3 Compatibility with Classical Causality

The full logical argument of how classical causality is derived from EPP is in the section "Causality as Effect Propagation Process". While classical causality can be derived from EPP, the reverse is not true because EPP cannot be derived from classical causality because classical causality requires a background spacetime. That deduction proves that EPP is more abstract in the sense of more general than classical causality. Therefore, it follows that EPP naturally applies to areas where classical causality cannot be used any longer. The internal validity of EPP roots in its internal soundness and consistency that stems from its first-principles reasoning. Therefore, ambiguity, contradictory claims, and unjustified leaps in logic are avoided to the extent it is possible. Minor mistakes might be possible and the author is open to suggestions to improve EPP further.

11.2 External validity

Establishing the boundaries of generalization as a proxy for external validity requires a delicate balance of realistically acknowledging what EPP can address versus avoiding overstating any particular capability. Related to external validity is always the possibility of an alternative interpretation of the underlying premises.

11.2.1 Alternative interpretations

There are several potential alternative interpretations of the premises underlying the Effect Propagation Process framework.

Russell was more right than acknowledged

One can take the position that, if Russell deemed causality as a relict of a bygone era, then why not openly ask to abandon causality altogether and focus solely on descriptive and correlation-based data science?

DARPA disagrees[98]:

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

The problem is not a simple choice between correlation and causation. The deeper issue, as contemporary philosophers like 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. This leads to tools that are inadequate for the intricate complexity of reality. To address this core critique, the EPP lifts causality into a contextual generalization required to model dynamic systems that no longer adhere to a fixed background spacetime. This includes systems with non-Euclidean geometries, multiple interacting contexts, and emergent causal structures, as found in domains like avionics. The question of falsifiability, therefore, applies not to the EPP framework itself, but to the specific, testable models that are constructed within it. A model of an avionics system built using the EPP is indeed falsifiable against simulation data; the framework itself simply provides the language for expressing that model.

The author argues, a similar shift towards a richer contextualization of advanced dynamic models needs to happen in the correlation-based methodologies of deep learning as well to build tools more suitable for today’s reality. A transfer of core EPP concepts, i.e., the contextual hypergraph, uniform Euclidean and non-Euclidean geometries, into the foundations of deep learning is welcomed by the author.

Pluralism of causal concepts

Instead of a unified framework like EPP, one can argue in favor of the existing pluralistic reality where multiple disjoint concepts of (computational) causality exist for different levels of causal analysis.

As stated before, EPP does not seek to replace classical causality and all tools that are built atop the classical definition of causality. Instead, EPP seeks to advance the core concept of causality to meet increasingly challenging demands. Due to the novelty of this foundational work, a single coherent framework is preferred until it becomes clear which parts may branch out and become more specialized domains over time.

11.2.2 Boundaries of Generalization

The main boundary to EPP comes down to preventing the application of any quantum principle to macroscopic systems.

EPP does not endorse nor attempt to apply any quantum principle to non-quantum systems. Instead, it takes inspiration from quantum gravity to advance causality. Furthermore, it is crucial to understand that EPP does not propose that macroscopic complex systems are quantum mechanical in the way they physically operate. Instead, EPP attempts to transfer concepts from quantum gravity to advance causality further to model advanced dynamic systems with complex feedback loops that are notoriously hard to model with conventional causality tools.

Another boundary relates to the lower limit of complexity that can be solved with EPP. For example, abandoning the cause-effect distinction via the causaloid is too radical for many applications. EPP remains compatible with classical causality and therefore can potentially be used where classical causality would apply. However, when a simpler classical causality methodology can solve the problem at hand, it should be preferred over any more complex methodology.

To prevent conceptual overreach, the author does not claim to have invented post-quantum causality even though its inherent independence of spacetime would suggest this view. Furthermore, the author does not claim that the Effect Propagation Process framework would solve open questions in quantum physics or quantum gravity. However, because of the early stage of quantum gravity research, it is far from settled how causality fits into the suggested conceptualization of emergent spacetime. Instead, the Effect Propagation Process, as a philosophical framework inspired by concepts of quantum gravity, is defined as a spacetime-agnostic continuous effect propagation process.

11.2.3 Method Selection Criteria: Classical Causality vs. EPP

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. The following decision matrix supports the assessment of when to use which methods. This matrix provides guidance on selecting an appropriate causal modeling approach based on the temporal and spatial complexity of the system under investigation.

Feature Assessed	System Characteristic	Classical Methods	EPP Methods
Temporal Complexity			
	Single time scale + linear progression	Sufficient	
	Multiple time scales OR non-linear temporal relationships	May struggle	Consider EPP
	Multiple time scales AND non-linear temporal relationships	Insufficient	EPP Required
Spatial Structure			
	Euclidean space with fixed coordinates	Sufficient	
	Non-Euclidean OR dynamic spatial relationships	Limited/Difficult	EPP Advantageous
	Non-Euclidean AND dynamic spatial relationships	Very Limited	EPP Required
Combined Complexity			
Scenario 1	Linear Time & Euclidean Space	Preferred	
Scenario 2	Non-Linear Time OR Complex Spatial	Limited/Difficult	Consider EPP
Scenario 3	Non-Linear Time AND Complex Spatial	Insufficient	EPP Required

Table 3: Causal Method Selection Matrix

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.

12 Future Work

The preceding chapters have established the Effect Propagation Process as a conceptual and formal framework for modeling dynamic causality. The framework's philosophical underpinnings have informed its implementation in DeepCausality. The framework's core principle, Higher-Order Emergence, provides a formal language for describing systems capable of recursively evolving their own causal and contextual structures.

The EPP, in its most advanced modalities, operates in a reality where the classical pillars of verification and trust are no longer guaranteed, as established in the Epistemology. However,the very power of this principle creates a set of three profound crisis, as foreshadowed in the metaphysics:

1. **The Crisis of Justification:** In a system where new causal rules and contexts are constantly emerging, the fixed principles needed for classical justification disappear.
2. **The Crisis of Truth:** In a system that co-evolves with its factual Context, the stable, external reality required for a correspondence theory of truth dissolves.
3. **The Crisis of Explainability:** It might not be possible any longer to explain the outcome because of the previous crisis of truth and the crisis of justification.

These crises are fundamental properties of higher-order emergence encoded in the EPP and thus demands a new class of ontological primitive for a normative framework that shifts the anchor from epistemology (what is true) to teleology (what is its purpose). The Effect Propagation Process therefore proposes two new, first-class ontological primitives:

- **The Teloid:** A computable unit of purpose. Functioning as a prospective guard of intent, a Teloid would be 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 Effect Ethos:** A framework for validating outcomes. Functioning as a retrospective validator, the Effect Ethos would assess the holistic, emergent state of the system after a reasoning cycle to ensure fundamental principles such as safety, fairness, or regulatory compliance have been upheld. The Effect Ethos would leverage the EPP's isomorphic design to construct a verifiable 'machine ethos' from simpler Teloid primitives, creating a composable and mechanistic ethical framework from first principles. Instead of external post-hoc analysis, the proposed Effect Ethos would become an integral part of the EPP and its implementation DeepCausality.

When combined, the Teloid and Effect Ethos, form a plausible architecture within which ethics becomes a computable and verifiable. The distinction between a prospective "Teloid" (guarding actions) and a retrospective "Effect Ethos" (validating outcomes) exists for a specific reason. A proposes action A proposed action, i.e., "shut down air-flow," can be 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. The Effect Ethos provides the necessary tools for this kind of holistic and balanced systemic validation.

The concepts of 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 since developing because these immense challenges clearly fall outside the scope of the presented EPP, but still warrant further consideration. While the formalization is 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. For the actual implementation, the EPP and its implementation DeepCausality, provides the staging ground because:

- Real-world safety problems are not confined to simple geometries. Avionics and robotics safety needs native support for non-Euclidean geometries.
- Ethics never occurs in a vacuum. Therefore, an Effect Ethos requires multi-contextual support.
- Holistic ethical outcomes are emergent properties, thus dynamic and emergent causality are necessary to capture these.

Furthermore, the proposed Effect Ethos represents a significant departure from connectionist models and offers a novel, verifiable approach to building intelligent systems that can reason not just about what is true, but also about what is right with respect to a formalized ethos.

The author acknowledges that realizing this vision, irrespective of any particular implementation, will require deep inquiry, probing questions, and different perspectives from a multitude of diverse stakeholders. The transparent and open-governance model of the DeepCausality project, hosted at the LF AI & Data Foundation, provides a vendor-neutral venue for facilitating this discussion.

13 Conclusion

The Effect Propagation Process framework rethinks causality as a continuous transfer of effects originating from a potentially non-spatiotemporal underlying structure.

The EPP is implemented in the DeepCausality[9] Rust library, which is a hypergeometric computational causality library that enables fast and deterministic context-aware causal reasoning across Euclidean and non-Euclidean spaces. This operational realization substantiates the EPP's core principles and demonstrates he EPP's applicability to real-world complex systems. DeepCausality finds its application in modeling dynamic control systems used in financial markets, advanced analytics, and complex control systems.

The framework navigates the challenges of spacetime agnosticism, aligns with the concept of emergence, accommodates indefinite causal order, and remains compatible with classical causality. Furthermore, the notion of an Effect Propagation Process reshapes the ontology of causality by suggesting that the most fundamental reality is not spacetime, but the generative process that materializes the spacetime context through which effects propagate.

The epistemology of the Effect Propagation Process reflects the complex systems it is designed to model by scaling with the modality of the EPP. For a static EPP, a positivist epistemology remains sufficient. For a dynamic EPP, the epistemology evolves towards an interpretivism perspective, and for an emergent EPP, a pragmatism perspective on the epistemology becomes necessary.

Likewise, for the justification of knowledge in an EPP, the underlying notion of truth scales with the modality of the EPP. For a static EPP, the meaning of truth aligns with the classical correspondence theory. However, in a dynamic EPP, the meaning of truth shifts towards a coherent adaptability approach. In an emergent EPP, the meaning of truth evolves towards pragmatic efficacy where the validity of relativistic, emergent causal relationships is established by their functional utility.

The framework provides a robust conceptual grounding for exploring the nature of causal effect propagation in a universe that may fundamentally defy classical intuition while leaving traditional teleology as a likely emergent property. It redefines the scope and methods of epistemology, shifting the focus of causal knowledge from observing event sequences in spacetime to inferring the rules and dynamics of this deeper process.

The Effect Propagation Process offers a unified philosophical language of causality that is powerful enough to handle new challenges, remains compatible with classical causality, and conceptually aligns with contemporary theories of quantum gravity. Future work may explores the realm of dynamic emergent causality further.

[9] `https://deepcausality.com`

References

[1] Stanford Encyclopedia of Philosophy. Plato's Timaeus. Stanford Encyclopedia of Philosophy, Online Resource. Accessed on May 18, 2025.

[2] Stanford Encyclopedia of Philosophy. Aristotle's Metaphysics. Stanford Encyclopedia of Philosophy, Online Resource. Accessed on May 18, 2025.

[3] Stanford Encyclopedia of Philosophy. Aristotle on Causality. Stanford Encyclopedia of Philosophy, Online Resource. Accessed on May 18, 2025.

[4] Lucius Annaeus Seneca. Seneca: Selected philosophical letters: Translated with introduction and commentary. Book, OUP Oxford, 2007.

[5] Stanford Encyclopedia of Philosophy. Leibniz's Philosophy of Physics. Stanford Encyclopedia of Philosophy, Online Resource. Accessed on May 18, 2025.

[6] Einstein Relatively Easy. Nov 1915 The four Einstein's papers. Website. Accessed on May 18, 2025.

[7] Bertrand Russell. On the notion of cause. Proceedings of the Aristotelian Society, Vol. 13, 1912. Pages 1–26.

[8] Michael E Peskin. *An Introduction to quantum field theory*. CRC press, 2018.

[9] Lucien Hardy. Probability theories with dynamic causal structure: a new framework for quantum gravity. arXiv preprint, 2005. arXiv:gr-qc/0509120.

[10] L. Mrini and L. Hardy. Indefinite causal structure and causal inequalities with time-symmetry. arXiv preprint, 2024. arXiv:2406.18489.

[11] Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv preprint arXiv:1801.04016*, 2018.

[12] Thomas S Verma. Causal networks: semantics and expressiveness. *Technical Report R-65, Cognitive Systems Laborator, University of California, Los Angeles*, 1986.

[13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.

[14] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

[15] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[16] Judea Pearl. *Causality*. Cambridge university press, 2009.

[17] J Pearl. Causality: models, reasoning, and inference cambridge university press new york. *NY, USA*, 2000.

[18] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT press, 2017.

[19] 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.

[20] 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.

[21] 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.

[22] Karthika Mohan and Judea Pearl. Graphical models for processing missing data. *Journal of the American Statistical Association*, 116(534):1023–1037, 2021.

[23] Thomas S Richardson and Peter Spirtes. Causal inference via ancestral graph models. *Oxford Statistical Science Series*, pages 83–105, 2003.

[24] Thomas S. Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 1990.

[25] 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.

[26] Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960, 1986.

[27] 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.

[28] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

[29] 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.

[30] 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.

[31] 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.

[32] Jin Tian and Judea Pearl. Probabilities of causation: Bounds and identification. *Annals of Mathematics and Artificial Intelligence*, 28(1):287–313, 2000.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] Jingwen Zhang, Yifang Chen, and Amandeep Singh. Causal bandits: Online decision-making in endogenous settings. arXiv preprint arXiv:2211.08649, 2022.

[38] Blair Bilodeau, Linbo Wang, and Daniel M. Roy. Adaptively exploiting d-separators with causal bandits, 2022.

[39] 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.

[40] Jeroen Berrevoets, Sam Verboven, and Wouter Verbeke. Treatment effect optimisation in dynamic environments. *Journal of Causal Inference*, 10(1):106–122, 2022.

[41] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

[42] Frederick Eberhardt. Introduction to the foundations of causal discovery. *International Journal of Data Science and Analytics*, 3(2):81–91, 2017.

[43] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

[44] 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.

[45] 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.

[46] Uzma Hasan, Emam Hossain, and Md Osman Gani. A survey on causal discovery methods for i.i.d. and time series data, 2024.

[47] Yinghua Gao, Li Shen, and Shu-Tao Xia. Dag-gan: Causal structure learning with generative adversarial nets. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3320–3324. IEEE, 2021.

[48] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[49] Clive WJ Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and control*, 2:329–352, 1980.

[50] John Geweke. Measurement of linear dependence and feedback between multiple time series. *Journal of the American statistical association*, 77(378):304–313, 1982.

[51] P Padav. Granger causality in time series–explained using chicken and egg problem, 2021.

[52] 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.

[53] 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.

[54] 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.

[55] N. Deng and Q. Zhang. Towards Dynamic Uncertain Causality Graphs for the Intelligent Diagnosis and Treatment of Hepatitis B. *Symmetry*, 12:1690, 2020.

[56] Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):19, 2019.

[57] 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.

[58] Jeroen Berrevoets, Krzysztof Kacprzyk, Zhaozhi Qian, and Mihaela van der Schaar. Navigating causal deep learning, 2022.

[59] Jeroen Berrevoets, Krzysztof Kacprzyk, Zhaozhi Qian, and Mihaela van der Schaar. Causal deep learning, 2024.

[60] 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.

[61] Zizhen Deng, Xiaolong Zheng, Hu Tian, and Daniel Dajun Zeng. Deep causal learning: representation, discovery and inference. *arXiv preprint arXiv:2211.03374*, 2022.

[62] Vikas Ramachandra. Deep learning for causal inference. *arXiv preprint arXiv:1803.00149*, 2018.

[63] Beatrice Acciaio, Anastasis Kratsios, and Gudmund Pammer. Designing universal causal deep learning models: The geometric (hyper) transformer. *Mathematical Finance*, 34(2):671–735, 2024.

[64] Lucien Hardy. Probability theories with dynamic causal structure: a new framework for quantum gravity. *arXiv preprint gr-qc/0509120*, 2005.

[65] 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.

[66] 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].

[67] 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.

[68] Xavier Ouvrard. HYPERGRAPHS: AN INTRODUCTION AND REVIEW. *arXiv preprint arXiv:2002.05014*, March 2020.

[69] Claude Berge. *Graphs and hypergraphs*, volume 7. North-Holland publishing company Amsterdam, 1973.

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] M. Zhang and Y. Chen. Link Prediction Based on Graph Neural Networks. In *Proc. Advances in Neural Information Process-ing Systems*, 2018.

[83] 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.

[84] 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.

[85] 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.

[86] 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.

[87] Amit Sharma and Emre Kiciman. Dowhy: An end-to-end library for causal inference. *arXiv preprint arXiv:2011.04216*, 2020.

[88] 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.

[89] Yang Zhao and Qing Liu. Causal ml: Python package for causal inference machine learning. *SoftwareX*, 21:101294, 2023.

[90] Peter Naur. Programming as theory building. *Microprocessing and microprogramming*, 15(5):253–261, 1985.

[91] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[92] Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American statistical Association*, 100(469):322–331, 2005.

[93] Paul Dagum, Adam Galper, and Eric Horvitz. Dynamic network models for forecasting. In *Uncertainty in artificial intelligence*, pages 41–48. Elsevier, 1992.

[94] Montgomery Furth et al. *Metaphysics: Zeta, Eta, Theta, Iota. Books VII-X.* Hackett Publishing, 1985.

[95] Marvin Hansen. The effect propagation process: A philosophical framework for post-quantum causality. `https://github.com/deepcausality-rs/papers`, June 2025. Accessed: June 2025.

[96] 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.

[97] 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.

[98] 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.

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 ($\{\text{true, false}\}$). (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 CausaloidGraph 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)