

EMET-LABS PROPOSAL TO HOST DEEPCAUSALITY IN LF AI & DATA

AUGUST 10, 2023

MARVIN HANSEN, DIRECTOR [EMET-LABS.COM](https://emet-labs.com)

MARVIN.HANSEN@EMET-LABS.COM

WHY CONTRIBUTE DEEPCAUSALALITY TO LINUX FOUNDATION

- Neutral holding ground
 - vendor-neutral, not for profit
- Open governance model
 - Transparent and open governance model
 - Instill trust in contributors and adopters in the management of the project
 - Neutral management of projects' assets by the foundation
- Growing community
 - Increase visibility of project through LF ecosystem
 - Increase contributors by converting new & existing users
 - Opportunities to collaborate with other projects

AGENDA

- 1) Problem
- 2) Challenge
- 3) How DeepCausality helps
- 4) Next steps

PROBLEM

WHEN DEEP LEARNING HITS ITS LIMITS

Emet Labs specializes in modelling volatility in financial markets

- Financial markets are increasingly interrelated
- Internal context (say month low/high) collides with external context (say Twitter)
- Complex spatial-temporal patterns in interconnected contextual time-series data

Deep Learning applied to market volatility falls short:

- Context-blind
 - World-model-blind
 - Data-relation-blind
- 
- 

ROOT CAUSE

- 1) Context-blindness roots in the universal approximation theorem
- 2) World-Model-blindness means DeepLearning does not have a model of the world that generates the data it uses to learn
- 3) Data-relational-blindness roots in the Independent and identically distributed (IID) data assumption

DARPA ACKNOWLEDGES THE PROBLEM:

”ANSR hypothesizes that several of the limitations in ML today are a consequence of the inability to incorporate contextual and background knowledge, and treating each data set as an independent, uncorrelated input. In the real world, observations are often correlated and a product of an underlying causal mechanism, which can be modeled and understood.”

Assured Neuro Symbolic Learning and Reasoning (ANSR) Program

[HTTPS://WWW.DARPA.MIL/PROGRAM/ASSURED-NEURO-SYMBOLIC-LEARNING-AND-REASONING](https://www.darpa.mil/program/assured-neuro-symbolic-learning-and-reasoning)

CHALLENGES

THREE BIG CHALLENGES

- 1) **Context & background knowledge**
- 2) **Relations between observations**
- 3) **Causal mechanism**

CONTEXT & BACKGROUND KNOWLEDGE

- Context & background knowledge is independent of the model
- Context data:
 - Differ over time i.e. today's report is different from last week's
 - Differ across time i.e. weekly reports contain different data than the annual report
 - Differ across space i.e. the Chicago office requires a different report than London
 - Differ across space and time: Zoom meeting across five different time zones...
- Context may need regular updates

RELATIONS BETWEEN OBSERVATIONS

- Temporal relations:
 - Last month high/low serves as reference point for the current data
 - One dimensional over T (Time)
 - Spatial relations:
 - Geometric patterns i.e. candle sticks relate to current data
 - Two or three dimensional, depending on the choice of representation
 - Temporal-spatial relations
 - Time varying geometric patterns that inform current observation in a progressing time continuum
 - Four-dimensional: 1D Time + 3D Space
- 
- 

CAUSAL MECHANISM (1 / 2)

- Correlation: IF A occurs, B occurs as well
 - Correlations are omnipresent i.e. rain and umbrella
 - Can be reversed i.e. umbrella and rain
- Causation: IF A then B; AND if NOT A then NOT B
 - Causation is an underlying structure:
 - IF power on, then lightbulb on; AND If the power is not on, the light bulb is not on
 - Cannot be reversed under the time arrow assumption i.e. the lightbulb cannot be on without its cause
- All causation can be expressed as correlation i.e. power on and light on
- Correlation cannot be expressed as causation. It's impossible in absence of a causal relation
- Finding causal relations and structures is fundamentally an exceptionally hard problem

CAUSAL MECHANISM (2/2)

- **Single cause:** These are extremely rare, but easy
 - Smoking cigarette at an open petrol tank -> Explosion
- **Multi-cause, single effect:** A bit more common
 - High unemployment, high interest rates, high inflation / costs of living (causes) lead to decreased disposable income (effect) resulting in lower real-estate sales (higher order effect).
- **Multi-cause, multiple effects:** Very common
 - High unemployment, low security standards, and poor governance may lead to vandalism and higher crime
- **Multiple-stages, multiple causes, and multiple effects**
 - Let's call this actual reality: Degree of complexity approximates infinity the more you dive into details
- **And causality is contextual over space, time, and space-time**

SOLUTION SPACE

- [Judea Pearl](#) at UCLA: Foundational work & structural causal models
- [Ilya Shpitser](#), Causal AI Lab at Johns Hopkins University: Causal and semi-parametric inference
- [Miguel Hernan](#), [Causal Lab](#) at Harvard University: Causal models applied to Infectious diseases, mental health, and veterans' health
- [Elias Bareinboim](#) at Columbia University: Causal inference with decision-making/reinforcement learning
- [Lucien Hardy](#) at the Perimeter Institute for theoretical physics: Causality foundation for Quantum Gravity
- [Causality and Machine Learning](#) at Microsoft Research: ALICE, Causica, DiCE, DoWhy, EconML
- [CausalML](#) at Uber: Causal inference with machine learning algorithms
- [Causal Inference Applications](#) at Netflix: Causal recommendation models & subscriber retention

OBSERVATIONS

- 1) Advanced research available
- 2) Microsoft and Uber at the forefront of industry adoption
- 3) Very diverse projects: Foundational work, Causal RL, EconML, Healthcare...
 - Impossible to compare projects in a feature matrix
- 4) Projects focus more on algorithm and application
- 5) Python is the lingua franca; Rust not yet explored
- 6) Contextualized causal inference not yet explored
- 7) Causal (hyper) geometric structures not yet explored

KEY FACTORS TO SUCCESS

1) Causal structure:

- Expresses arbitrary complex causal relations between causes, data, and context

2) Context:

- Relates causal relations to contextual data

3) Explanation

- Causal reasoning with explanation helps to understand the process

DEEPCAUSALITY

HOW DOES DEEP CAUSALITY DIFFER FROM DEEP LEARNING?

DEEP CAUSALITY

- Free of the IID assumption & explicit assumptions
- Deterministic causality
- Specializes
- Explainable
- Context aware
- Content with little data
- Small model of causal relations
- Good for reasoning, control systems & anomaly detection

DEEP LEARNING

- Subject to the IID assumption & implicit assumptions
- Non-deterministic correlation
- Generalizes
- Hard to explain; possible but needs work
- Context-free
- Requires big data
- Big model
- Good for generation, classification & language models

NOT ALL CAUSAL PROBLEMS ARE CREATED EQUAL

Simple vs. Complex	Static vs. Dynamic	Deterministic vs Probabilistic
Simple problems need simple solutions with simple structures.	Static problems need efficient representation guaranteed to be invariant.	Deterministic problems require verification and explanation.
Complex problems need complex structures and techniques to manage complexity.	Dynamic problems need an efficient update mechanism.	Probabilistic problems require a simple integration into the causal world.

FOUR LAYERS OF CAUSAL REASONING



Observations: Data we observe in the world



Assumptions: What we assume about the data we observe



Inference: What we can infer from the data with those assumptions

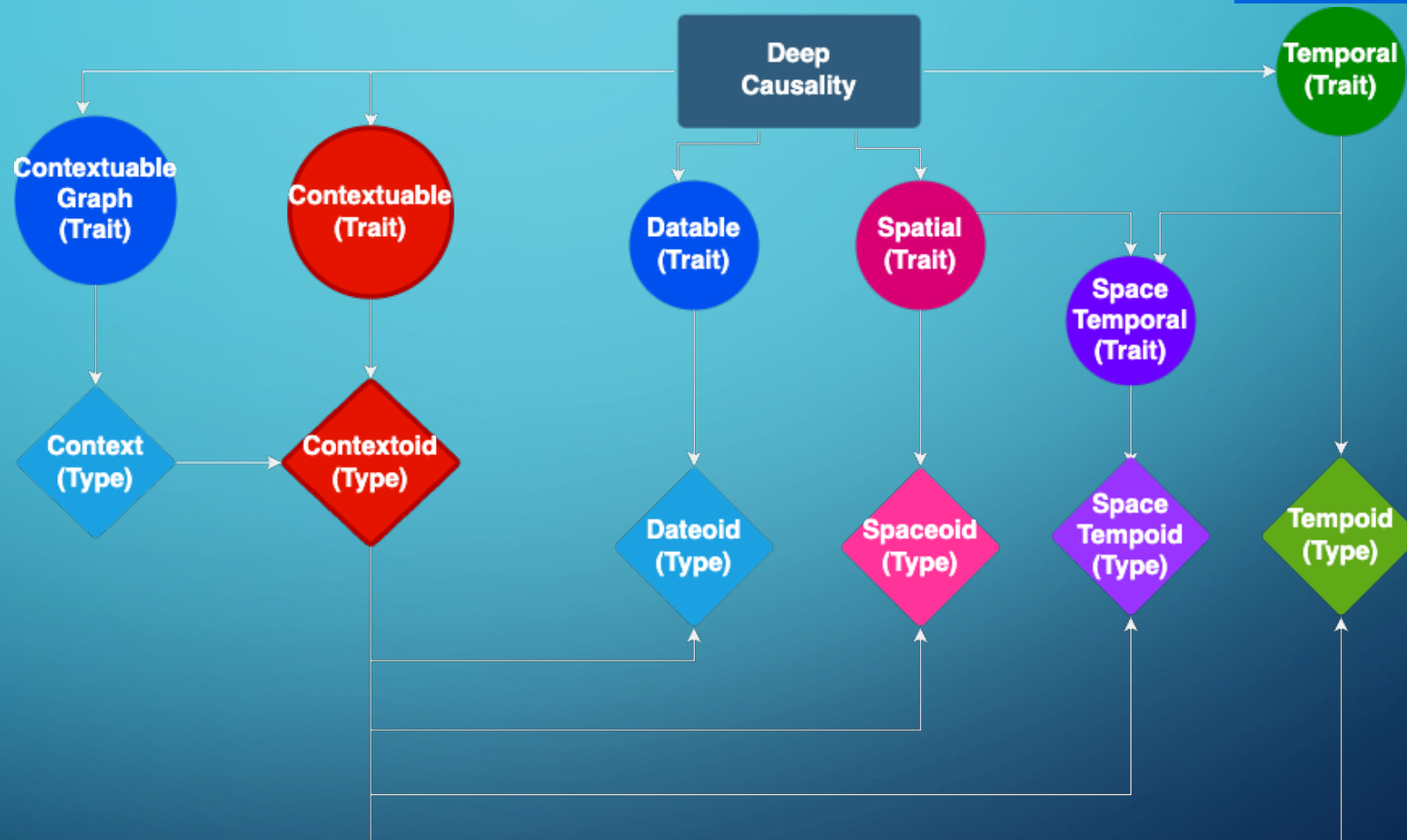


Causality: The causal relationships that structure inferences

ARCHITECTURE (EXCL CONTEXT)



CONTEXT ARCHITECTURE



IMPLEMENTATION PRINCIPLES

- Protocol based causality
 - Inspired by differential programming pioneered by Google research in 2020, adopted by PassiveLogic
 - Defines traits with default implementation (Think protocols and extensions in Swift)
 - Extends standard collections in extension traits with functionality from a generic default implementation
- Recursive isomorphic data structures
 - Recursion in Rust only requires a layer of indirection i.e. a reference
 - Isomorphism requires that all data entities share the same structure: Hence Causaloid & Contextoid
- Disjoint algebraic data types a.k.a nested Enum
 - Solves the problem of missing inheritance & shared supertype
 - Nests various traits in an enum and then wraps the enum in a struct



FOUR PILLARS OF DEEP CAUSALITY

- 1) **Hypergeometric deep causality**
- 2) **Recursive causal data structures**
- 3) **Hypergraph context**
- 4) **Causal state machines**

HYPER-GEOMETRIC COMPUTATIONAL CAUSALITY



HYPER-GEOMETRIC COMPUTATIONAL CAUSALITY

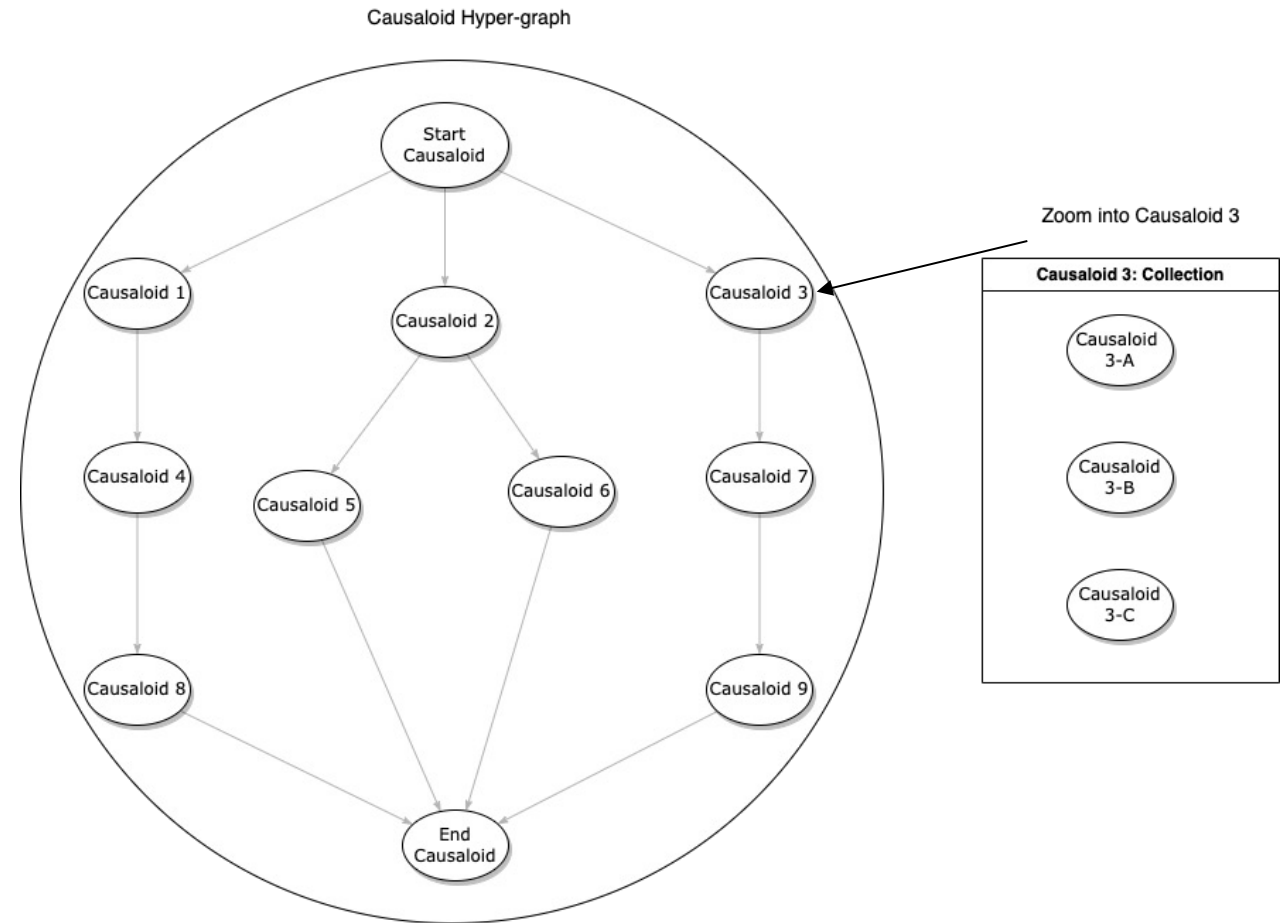
- **More than one way to represent causal structure: Algebra vs Geometry**
 - **Algebra: Complex arithmetic**
 - **Geometry: Complex structure**
- **DeepCausality choses a hypergraph geometric representation**
- **DeepCausality solves complexity with recursive causal data structures**

RECURSIVE CAUSAL DATA STRUCTURES

RECURSIVE CAUSAL DATA STRUCTURES

- Causaloid central structure
- Causaloid: Concept borrowed from Lucien Hardy's work on Quantum Gravity
- Causaloid defines a causal relation as a causal function
- Causal collections scale from simple to complex: Vector, Map, Hypergraph
- The causal graph is a hypergraph comprising of causaloids
 - A collection stores multiple causaloids, the collection is stored in a causaloid, and the causaloid may then be stored in a hypergraph, which itself is stored as a causaloid
 - Causal reasoning across the entire graph, selected nodes, selected path, or shortest path

RECURSIVE CAUSAL DATA STRUCTURES



IMPLEMENTATION

- Causalgraph builds upon a sparse matrix hypergraph from the petgraph lib
- Trait defines behavior relative to the causable trait and its causaloid impl.
- Causal collection implemented as extension trait
- Causaloid uses an internal type Enum to determine if it's a singleton, graph, or a collection
 - Depending on the internal type, different inference functions are called
 - However, the last element must be singleton to let the recursion terminate eventually
- Causaloid uses an internal bool flag to determine if it has a context
 - Depending on the flag, it either calls a causal function with context or one without context
 - Constructor sets bool flag, type Enum, and context.

HYPERGRAPH CONTEXT

HYPERGRAPH CONTEXT

- Time:
 - Conventionally seen as continuous discrete series: $T_0, T_1, \dots T_n$
- Space:
 - Conventionally seen as static location i.e. coordinates that are time invariant
- Space-Time:
 - Conventionally not much considered outside theoretical physics and space industry
- Data context:
 - Might be time varying, space-varying, or even space-time varying

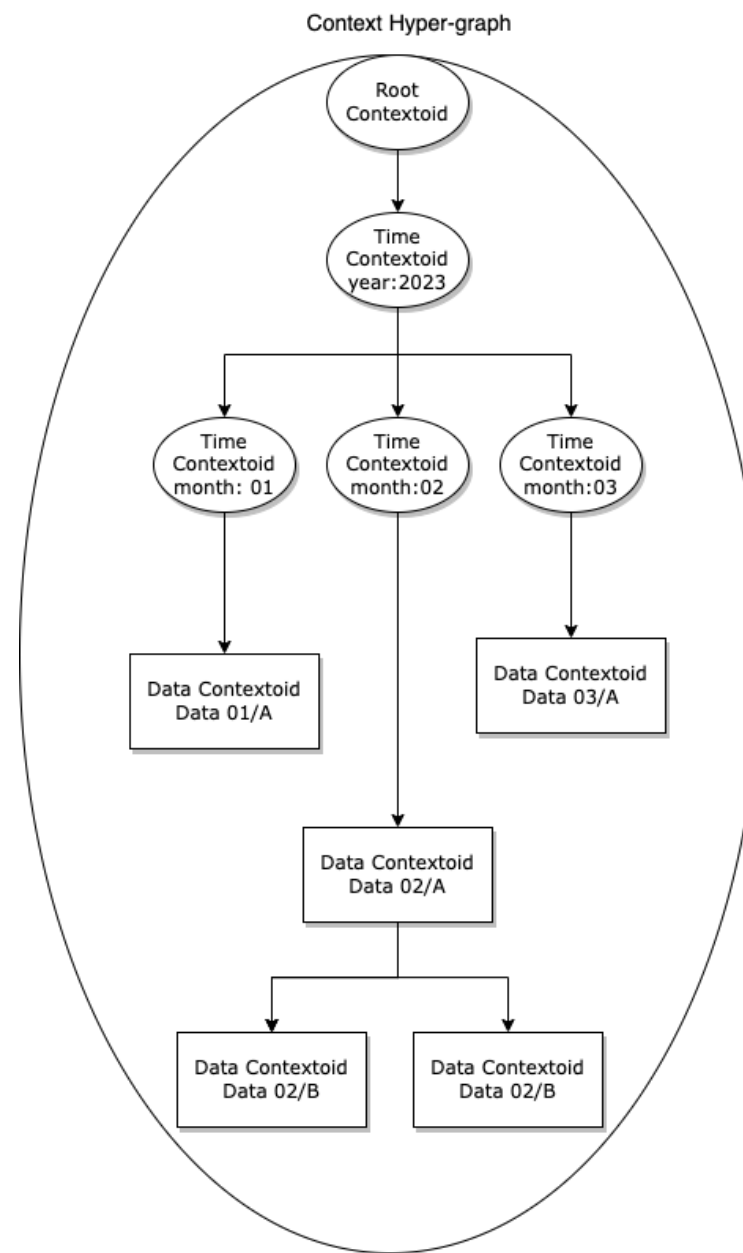
HYPERGRAPH CONTEXT IN DEEPCAUSALITY

- **Time:**
 - Considered as temporal hypergraph comprising of tempoids
 - Tempoid is one unit of time with a scale: Month: 12 (December)
- **Space:**
 - Considered as spatial hypergraph comprising of spaceoids
 - Spaceoid is one unit of space with coordinates
- **Space-Time:**
 - Considered as spacetime hypergraph comprising of spacetempoids
 - Spacetempoid is one unit of spacetime with coordinates at one unit of time
- **Data:**
 - Might be anything depending on how you define your data

HYPERGRAPH CONTEXT IN DEEPCAUSALITY

- Contextoid is the central structure
- Contextoids can be data-like, time-like, space-like, or any combination
- Contextoids might be referenced by one or more causaloids
- Context hypergraph
 - Graph of contextoids that represent, data, time, space, or space-time
 - A contextoid may link to any number of other contextoids

HYPERGRAPH CONTEXT

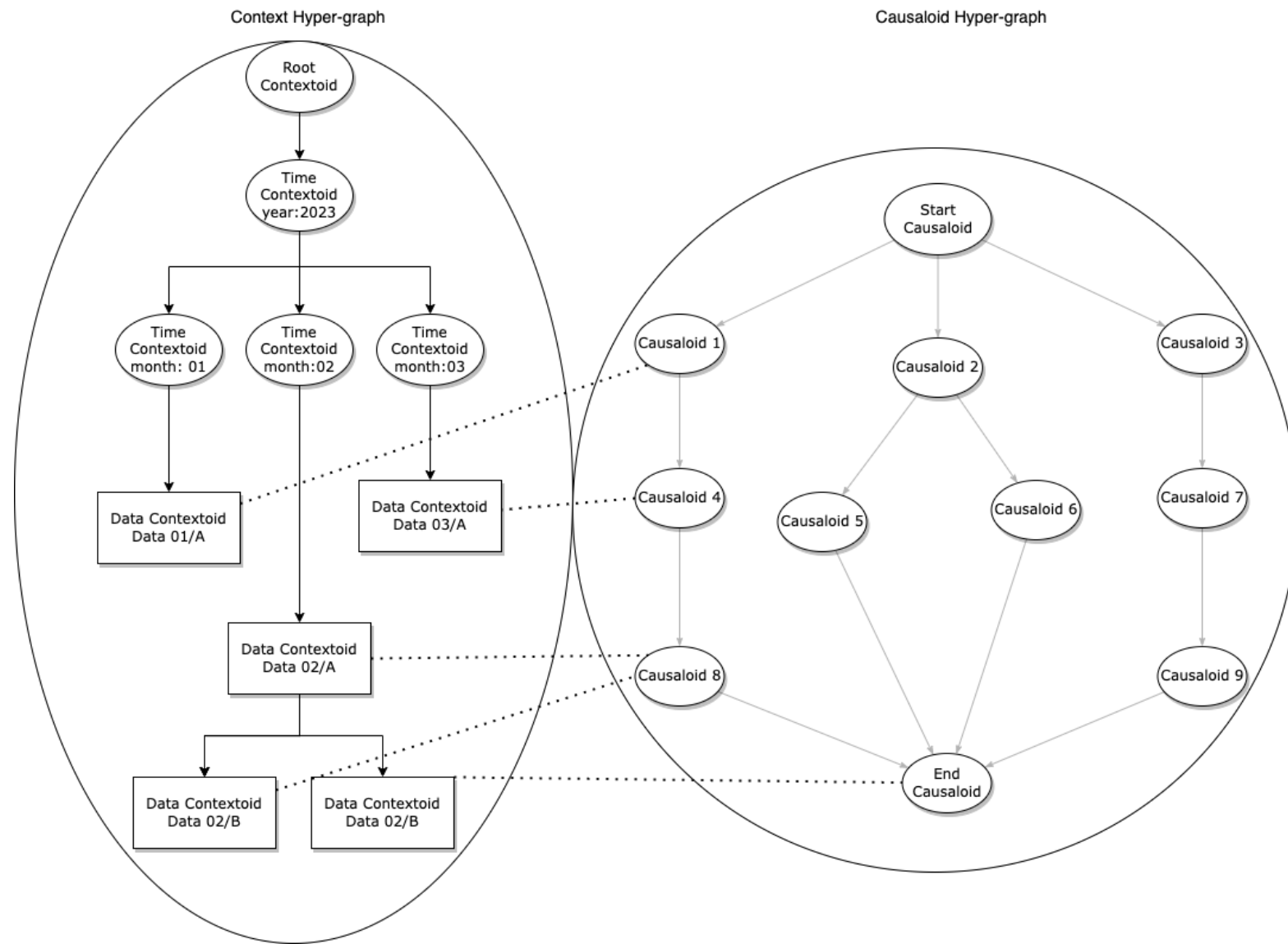


IMPLEMENTATION

- Like the causal structure, context builds upon a sparse matrix hypergraph
- Unlike the causal structure, context is not recursive
- Traits serve more as an interface for both, the contextoid and the context
- Contextoid wraps disjoint algebraic data structure a.k.a nested Enum
 - Each Enum embeds a type trait i.e. Temporal to allow custom types
- Custom types must extend the corresponding (super) trait i.e. temporal and implement both, the custom trait and the super trait

CAUSALOID + CONTEXT
=
CAUSAL MODEL

CONTEXTUALIZED CAUSAL MODELS



IMPLEMENTATION

- The context comes in two variants: `BaseContext` and `Context`.
 - `BaseContext` is an alias to the default (basic node type) that removes generic type annotations
 - `Context` requires to overwriting each node type with either a default or a custom type
 - In practice, only few custom types are needed, and default types fill the rest
- The causaloid must be constructed with context and with `ContextCausalFunction`
 - Functionality of custom types is only accessible inside the context function when the custom trait has been imported
 - The Causaloid dispatches automatically between functions with or without context.
- BTW, in `DeepCausality` it's all static dispatching so no dynamic overhead

CAUSAL STATE MACHINES



CAUSAL STATE MACHINES (CSM)

- Problem: Finite State Machines (FSM) require a priori knowledge of all states
- Solution: Causal State machine (CSM) generalizes over FSM and allows dynamic state management
- CSM
 - Causal State = Causaloid with its causal function
 - Causal Action = The effect function triggered when cause was detected
 - Context-free = State machine can only work with defined states.
- Implemented via safe function references (pointers) in Rust
- CSM can be generated, evaluated, and executed on-demand
- Ideal for dynamic control systems

IMPLEMENTATION

- Uses the BaseCausaloid since neither customization nor context is needed
- Adds two more types:
 - CausalState: defines a target state
 - CausalAction: Defines what to do when the target state has been reached
- CasualStateMachine
 - Evaluates if the target state has been reached and then triggers the action
- Can dynamically add, remove, or evaluate states at runtime

CAUSALOID VS. CAUSAL STATE MACHINES

Causaloid

- Contextualized
- Dynamic w.r.t. model & context
- Flexible: Can be probabilistic, deterministic, or both.
- Action defined outside the model

Causal State Machine

- Context-free
- Static
- Strictly deterministic
- Defined causal action

THE VALUE OF DEEPCAUSALITY

VALUE

- 1) Comprehensive data enrichment
- 2) Efficient causal representation
- 3) Gives reason across the chain of causality
- 4) Inspires new directions i.e. contextualized deep learning

DEEPCAUSALITY FEATURE

- 1) Context
- 2) Causaloid data structure
- 3) Explainability
- 4) Exploration of new ideas, data structures, and implementation techniques

NEXT STEPS

NEXT STEPS

Roadmap:

- Support multiple contexts
 - Complex models may require more than one context
- Explore causal learning
 - Right now, causal models are crafted by hand
 - Groundwork has been laid via assumable & inferable protocol
 - However, existing work of causal RL learning does not transfer well
 - Deep Neuroevolution might offer a novel path towards causal learning
- Expand docs & code examples

INFORMATION FOR PROPOSAL

- License: MIT
- GH repo: <https://github.com/deepcausality-rs>
- Proposal: <https://github.com/lfai/proposing-projects/blob/master/proposals/deepcausality.adoc>
- Possible Collaboration in LF AI & Data:
 - 1chipML: Machine learning for microcontrollers
 - ForestFlow: Machine learning model server
 - Xtreme1: ML platform for multisensory training data
 - BentoML: The Unified AI Application Framework

Thank You

Marvin Hansen

marvin.hansen@emet-labs.com