Rethinking AI: From Functions to Functors

Sridhar Mahadevan

Adobe Research
smahadev@adobe.com

Abstract

We propose a new theoretical foundation for artificial intelligence (AI) and machine learning (ML), building on ideas in pure mathematics relating to categories and functors. This paper builds on our AAAI 2025 tutorial *Thinking with Functors: Category Theory for A(G)I*, which provides background material. In addition, our recent papers on *intuitionistic j-do calculus in Topos Causal Models* and *GAIA: Categorical Foundations of Generative AI* illustrate how to generalize well-known formalisms in AI, such as causal inference and deep learning, to a category-theoretic setting.

AAAI 2025 Tutorial: Thinking with Functors — https://people.cs.umass.edu/~mahadeva/papers/aaai2025-tutorial-th18.pdf

Intuitionistic j-do-calculus in Topos Causal Models — https://arxiv.org/abs/2510.17944

GAIA: Categorical Foundations of Generative AI — https://arxiv.org/abs/2402.18732

Introduction

In this paper, we propose a novel theoretical framework for AI and ML based on the 21st century mathematical framework of categories and functors (Riehl 2017; Richter 2020). Much of the past six decades of research in AI and ML is based on 17th and 18th century mathematics: calculus, set theory, graphs, and probability. While these well-developed formalisms have been invaluable in developing AI, recent advances require a more sophisticated abstract framework that is intrinsically *compositional* (Fong and Spivak 2018). Our proposed framework is built around *functors*, which are mappings that transform an input *category* into an output *category* (see Figure 1). A functor is required not only to map elements of the input space (or category) into an output space (or category), but it is also required to preserve morphisms (or arrows) between elements. In this paper, we give a range of current applications of functor-based approaches to AI and ML, and also include a brief deeper dive into the main concepts of category theory.

Additional motivation for considering functor-based methods arises from the growing realization that current function approximation based methods are reaching a hard limit on their performance. While ML frameworks, such as deep learning (Bengio 2009), have ushered in remarkable advances in the ability to train large foundation models (DeepSeek-AI 2024), a growing consensus is emerging in the AI research community that despite being trained on nearly all of humanity's digital footprint, existing foundation models have significant limitations. Some of these limitations are a result of an inability of Transformer models (Vaswani et al. 2017) to form compositionally correct generalizations (Dziri et al. 2023). Vafa et al. (2025) have shown recently that a foundation model trained on planetary orbits was able to predict extremely well, but lacked the central insight of Newton that Kepler's elliptical trajectories are caused by an inward force directed to the sun, demonstrating a fundamental lack of understanding of gravity. Pearl and Mackenzie (2018) have proposed the *Ladder of Causality*, where ML methods such as deep learning used to build foundation models are restricted to the bottom-most layer, and higher level layers require experimentation or counterfactual reasoning, and cannot emerge from a purely bottom-up inductive process.

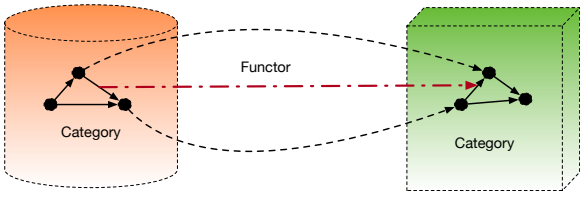


Figure 1: Categories are collections of objects, with a collection of arrows defined between each pair. Functors map between categories, and need to transform both objects and arrows from a domain category into a co-domain category.

Categories and Functors

Category theory (MacLane 1971; Riehl 2017; Lurie 2009) represents the most significant unification of mathematics since antiquity. A *category* is simply a collection of *objects* that interact pairwise with each other through a set of *arrows*. Much of the richness of applications of category theory is a consequence of its abstractness: objects can be common mathematical structures – groups, rings,

modules and vector spaces, topological spaces or measurable spaces – but more interestingly, they can be neural networks (Fong, Spivak, and Tuyéras 2019), causal models (Mahadevan 2023), probability distributions (Fritz and Klingler 2023), or Markov decision processes (Mahadevan 2021). Arrows are defined in terms of the category structure, so for groups, arrows are group homomorphisms, and for measurable spaces, arrows are measurable functions. For MDPs, arrows define MDP homomorphisms (Rezaei-Shoshtari et al. 2022). Categories can be mapped from one to another through *functors*. A functor comprises of two parts: an *object function* that maps each object in the domain category to another object in the co-domain category; and an *arrow function* that maps each arrow from the domain to the co-domain category. Two functors that map between the same two categories can be related through *natural transformations*.

To give a concrete example, we can define clustering formally as a functor F that maps the input category of finite metric spaces **FinMet** defined by (X, d), where X is a finite set of points in $\mathbb{R}^n$ and $d : X \times X \rightarrow [0, \infty]$ is a (generalized) finite metric space, and the output category **Part** is the set of all partitions X into subsets X_i such that $\cup_i X_i = X$.

Category theory is a *compositional theory*: one can construct a category in a myriad ways, ranging from mathematical structures such as groups, rings, vector spaces, and measurable (probability) spaces, to categories over large language models (LLMs) (Bradley, Terilla, and Vlassopoulos 2022) and categories over causal models (Mahadevan 2023). Functors have wide applicability to AI. Fong, Spivak, and Tuyéras (2019) show how to formulate backpropagation in deep learning as a functor that maps from the category of graphs representing structure to the category of learners, where each object defines a compositional learner. UMAP is arguably the most successful data visualization method in ML today, and it is based on constructing functors over *simplicial sets*, a type of higher-order category theory (McInnes, Healy, and Melville 2018). The primary role of this paper and presentation is to illustrate the idea that reformulating ML from approximating functions to *extending functors* might usher in transformative insights that could lead to the next set of practical breakthroughs. In addition, the massive energy costs of deep-learning based approaches is motivating significant research into quantum computation. An elegant way to understand quantum computing is to view it in terms of *string diagrams* in category theory (Heunen and Vicary 2019).

Functors in AI and ML

Every analogy is yearning to be a functor – John Baez

Our proposed framework is based on extending functors, rather than approximating functions from data. To understand the conceptual shift in focus from set-theoretic functions to category-theoretic functors, we will briefly illustrate the wide variety of functors that are of relevance to AI and ML (see Table 1). We briefly explain each example in more detail below, giving suitable references to published literature.

Functor	Domain	Co-domain
Clustering	Finite metric spaces	Partitions
Backpropagation	Graphs	Learners
Causality	Monoidal cat.	Markov chain
LLM	Text	Probability
RL	Markov decision process	Policies

Table 1: Illustrating the diverse applications of functors in AI and ML: each functor is defined over a domain and co-domain category.

- *Clustering as a functor*: Carlsson and Mémoli (2013) formulated the classic problem of clustering as learning a functor from the domain category of finite metric spaces to the output category of partitions (sets), thereby resolving an impossibility theorem stated by Kleinberg earlier.
- *Backprop as a functor*: Fong, Spivak, and Tuyéras (2019) showed how backpropagation – the workhorse algorithm underlying deep learning – can be modeled as a functor that maps from the category of graph structures, representing the syntax of a neural network, to the category of *Learners* where each object defines a building block of a supervised ML function approximation module.
- *Causality as a functor*: A growing literature has studied the application of category theory to causal inference (Fong 2012; Jacobs, Kissinger, and Zanasi 2018; Mahadevan 2023). Any causal model that is defined as a structural causal model (SCM) (Pearl 2009) can be encoded as a functor mapping a symmetric monoidal category representing the structure of the SCM to the category of stochastic processes. Fritz (2020) defines *Markov categories* and shows how they define an overarching category-theoretic language expressive enough to formulate a large variety of problems in probability theory and statistics.
- *LLM as categories:* Bradley, Terilla, and Vlassopoulos (2022) define an *enriched* category theory for LLMs, which encode both the syntax of natural language sentences, as well as its semantics. Here, objects represent fragments of language, such as "I am flying", and arrows define completion probabilities to extensions such as "I am flying to Singapore to attend AAAI 2026".
- *RL as a category*: Mahadevan (2021) defines RL as a category over objects defined as Markov decision processes (MDPs) (Bertsekas 2019), where the arrows define homomorphisms between MDPs that abstract the state, action, and reward functions in a optimality-preserving manner. This formulation easily extends to other models, such as predictive state representations (PSRs) or linear dynamical systems.

Why Clustering Should be a Functor

A central notion that underlies our framework is to think of AI and ML in terms of functors, not functions. A remarkably simple and illustrative example comes from clustering, one of the most well-studied problems in ML and statistics. Kleinberg (2002) showed that one can impose three criteria on a clustering algorithm, which seem entirely natural,

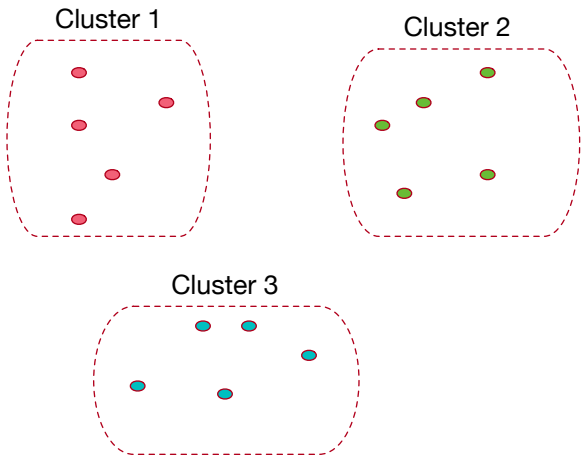


Figure 2: Clustering is one of the most traditional problems in statistics and ML, which requires constructing a partition of a finite metric space by grouping points together based on their pairwise distances. Kleinberg (2002) proved that no clustering algorithm exists that satisfies three simple properties. Carlsson and Mémoli (2013) showed that treating clustering as a functor resolves the impossibility result, and leads to a constructive algorithm.

and remarkably, no standard clustering algorithm satisfies all these conditions:

- *Scale invariance:* If the distance metric d is increased or decreased by $c \cdot d$, where c is a scalar real number, the output clustering should not change. In terms of Figure 2, if the points in each cluster became closer together or further apart proportionally, the clustering should remain the same.
- *Completeness:* For any given partition of the space X, there should exist some distance function d such that the clustering algorithm when given that distance function should return the desired partition.
- *Monotonicity:* If the distance between points within each cluster in Figure 2 were decreased, and the distances between points in different clusters were increased, the clustering should not change either.

Carlsson and Mémoli (2013) show that modeling clustering as a functor resolves this impossibility theorem.

Causal Inference

In recent years, there has been significant interest in categorical models of causality, based on symmetric monoidal categories (Fong 2012; Fritz and Klingler 2023; Cho and Jacobs 2019; Jacobs, Kissinger, and Zanasi 2018), as well as simplicial sets and higher-order categories (Mahadevan 2023). Markov categories (Fritz 2020) define a broad unifying framework for probabilistic inference and statistics using symmetric monoidal categories, where each object is additionally equipped with a comonoidal "copy-delete" operation. It enables carrying out rigorous proofs using an elegant string diagrammatic language (Selinger 2010). Any causal model based on graphs (Pearl 2009; Forré and Mooij 2017; Spirtes, Glymour, and Scheines 2000) or other algebraic formalisms, such as integer-valued multisets (Studeny 2010), can be translated into a string diagram over a symmetric monoidal category, or a simplicial set. Operations on causal models, such as interventions, can be modeled as functors on the objects of the associated symmetric monoidal category or simplicial set. Categorical approaches to causality also extend to the *potential outcomes* counterfactual framework (Imbens and Rubin 2015).

Categorical approaches fundamentally differ from past work in causality in their focus on the elucidation of universal properties. In our previous work (Mahadevan 2023, 2025b), we introduced the framework of *universal causality* based on the notion of universal properties in category theory (Riehl 2017): a causal property is universal if it can be defined in terms of an *initial* or *final* object in a category of causal diagrams, or in terms of a *causal representable functor* using the Yoneda Lemma. For example, a structural causal model (SCM) (Pearl 2009) is defined as a (deterministic) mapping from a collection of exogenous variables into a collection of endogenous variables, derived by "collating" local functions that serve as independent causal mechanisms (Galles and Pearl 1988; Parascandolo et al. 2017). However, SCMs can be further analyzed in terms of their universal properties, such as categorical product, coproduct, limits and colimits, equalizers and coequalizers etc. These latter properties can be shown formally to be initial or final objects in a category of diagrams (Riehl 2017), or as representable functors through the Yoneda Lemma (MacLane 1971).

We have recently developed a a topos-theoretic framework for universal causality (Mahadevan 2025e), which leads also to an intuitionistic generalization of Pearl's do-calculus termed j-stable causal inference based on the Lawvere-Tierney topology on a topos defined by a modal operator j on the subobject classifier Ω (Mahadevan 2025c). In this paper, we define an intuitionistic logic called j-do-calculus (aka "judo calculus"), where we replace global truth with local truth defined by Kripke-Joyal semantics. We have recently completed an experimental study of a decentralized causal discovery framework based on judo calculus that implements j-do-calculus with well-known causal discovery procedures, including score-based, constraint-based and gradient based methods (Mahadevan 2025a). This experimental study of judo calculus shows how to (i) form data-driven j-covers (via regime/section constructions), (ii) compute chartwise conditional independences after graph surgeries, and (iii) glue them to certify the premises of the j-do rules in practice.

Judo calculus is based on topos theory (Bell 1988) a branch of category theory (MacLane 1971), which allows modeling causality in a more flexible way than classical methods. For example, in many real-world applications of causal inference, a particular intervention, such as administering a drug or employing a lunch program in schools, may not be effective over the entire population, but rather in different "regimes" (e.g., senior citizens may respond more favorably than younger recipients, and similarly, students from low-income backgrounds may benefit nutrition-

Characteristic	Judo Calculus
Logic	Intuitionistic logic: truth is *local*
Context	Local causal truth is "glued" together
Interventions	Subobject classifier
Identification	Judo calculus axiomatic framework

Table 2: Some of the salient features of judo calculus.

ally from a free lunch program than students from high-income regions). Table 2 summarizes the differences between classical do-calculus and judo calculus.

As a concrete example, let us imagine that a city planning a public health initiative to combat childhood obesity decides to distribute free healthy lunches in public schools. The city policy makers want to determine if this intervention reduces the students' body mass index (BMI) by the year's end. Classical do-calculus would seek a global average treatment effect over the entire school population. Judo calculus makes it possible to define individual regimes, such as "low-income" and "high-income" where the causal intervention may or may not be as effective.

Judo calculus works in the setting of sheaves or sites, which are categories that are equipped with a Lawvere-Tierney topology defined by a modal operator j on the subobjects. In plain English, this means that the category has a topology defined by the arrows, and their compositional structure. This j operator defines a notion of causal "stability", which will be studied extensively in the coming sections. For example, in the "low income" students, the city's causal intervention may be j-stable, whereas in the "high income" category, this intervention may not be as effective. The operator j acts on the subobject classifier Ω: an object in the category that serves to define "truth", which in general is not Boolean. The j operator acts to determine "local" truth from "global" truth, and also provides a closure property to determine j-stability. The j operator on Ω is defined to represent the notion of a "globally valid" causal statement. A j-cover represents a tesselation of the space of arrows that all have a common co-domain. Figure 3 summarizes the three rules of judo calculus.

A Deeper Dive into Category Theory

To understand the proposed framework of functors and their extensions, we need to do a deeper dive into category theory. We will focus the remainder of the paper on explaining the essential ideas. A simple way to understand the definition of a category is to view it as a "generalized" graph, where there is no limitation on the number of vertices, or the number of edges between any given pair of vertices. Each vertex defines an object in a category, and each edge is associated with a morphism. The underlying graph induces a "free" category where we consider all possible paths between pairs of vertices (including self-loops) as the set of morphisms between them. In the reverse direction, given a category, we can define a "forgetful" functor that extracts the underlying graph from the category, forgetting the composition rule.

Definition 1. *A* **graph** $\mathcal{G}$ *(sometimes referred to as a quiver) is a labeled directed multi-graph defined by a set O of* objects*, a set A of* arrows*, along with two morphisms $s : A \to O$ and $t : A \to O$ that specify the domain and co-domain of each arrow. In this graph, we define the set of composable pairs of arrows by the set*

$$A \times_O A = \{\langle g, f\rangle |\ g, f \in A,\ \ s(g) = t(f)\}$$

A **category** $\mathcal{C}$ *is a graph $\mathcal{G}$ with two additional functions: $\mathbf{id} : O \to A$, mapping each object $c \in C$ to an arrow $\mathbf{id}_c$ and $\circ : A \times_O A \to A$, mapping each pair of composable morphisms $\langle f, g\rangle$ to their composition $g \circ f$.*

1. **j-ignorability / j-elimination. If** $Y \perp\!\!\!\perp X \mid (W, J)$ ***on the cover*****, then**

$$\mathbb{P}_J(y \mid \mathrm{do}_J(x), w) \;=\; \mathbb{P}_J(y \mid x, w) \;=\; \mathbb{P}_J(y \mid w).$$

2. **j-action/observation exchange. If** $Y \perp\!\!\!\perp Z \mid (X, W, J)$ ***on the cover*****, then**

$$\mathbb{P}_J(y \mid \mathrm{do}_J(x), z, w) \;=\; \mathbb{P}_J(y \mid x, z, w).$$

3. **j-backdoor (adjustment). If Z is a J-admissible adjustment set for $X \to Y$ (i.e., blocks all J-backdoor paths), then**

$$\mathbb{P}_J(y \mid \mathrm{do}_J(x)) \;=\; \sum_z \mathbb{P}_J(y \mid x, z)\, \mathbb{P}_J(z).$$

Figure 3: The Three Rules of Judo Calculus.

Natural Transformations

We begin with the crucial concept of *natural transformations*. Given any two functors $F : C \to D$ and $G : C \to D$ between the same pair of categories, we can define a mapping between F and G that is referred to as a natural transformation. These are defined through a collection of mappings, one for each object c of C, thereby defining a morphism in D for each object in C.

Definition 2. *Given categories C and D, and functors $F, G : C \to D$, a* **natural transformation** $\alpha : F \Rightarrow G$ *is defined by the following data:*

- *an arrow $\alpha_c : Fc \to Gc$ in D for each object $c \in C$, which together define the components of the natural transformation.*
- *For each morphism $f : c \to c'$, the following commutative diagram holds true:*

$$\begin{array}{ccc} Fc & \xrightarrow{\alpha_c} & Gc \\ {\scriptstyle Ff}\downarrow & & \downarrow{\scriptstyle Gf} \\ Fc' & \xrightarrow[\alpha_{c'}]{} & Gc' \end{array}$$

A **natural isomorphism** *is a natural transformation $\alpha : F \Rightarrow G$ in which every component α_c is an isomorphism.*

This process of going from a category to its underlying directed graph embodies a fundamental universal construction in category theory, called the *universal arrow*. It lies at the heart of many useful results, principally the Yoneda lemma that shows how object identity itself emerges from the structure of morphisms that lead into (or out of) it.

Definition 3. *Given a functor $S : D \to C$ between two categories, and an object c of category C, a* **universal arrow** *from c to S is a pair $\langle r, u \rangle$, where r is an object of D and $u : c \to Sr$ is an arrow of C, such that the following universal property holds true:*

- *For every pair $\langle d, f \rangle$ with d an object of D and $f : c \to Sd$ an arrow of C, there is a unique arrow $f' : r \to d$ of D with $Sf' \circ u = f$.*

Definition 4. *If D is a category and $H : D \to \mathbf{Set}$ is a set-valued functor, a* **universal element** *associated with the functor H is a pair $\langle r, e \rangle$ consisting of an object $r \in D$ and an element $e \in Hr$ such that for every pair $\langle d, x \rangle$ with $x \in Hd$, there is a unique arrow $f : r \to d$ of D such that $(Hf)e = x$.*

Example 1. *Let E be an equivalence relation on a set S, and consider the quotient set S/E of equivalence classes, where $p : S \to S/E$ sends each element $s \in S$ into its corresponding equivalence class. The set of equivalence classes S/E has the property that any function $f : S \to X$ that respects the equivalence relation can be written as $fs = fs'$ whenever $s \sim_E s'$, that is, $f = f' \circ p$, where the unique function $f' : S/E \to X$. Thus, $\langle S/E, p \rangle$ is a universal element for the functor H.*

The Yoneda Lemma: Mapping Objects into Functors

Natural transformations lead to the most striking result in category theory: the Yoneda Lemma (MacLane 1971), which asserts that objects can be defined (upto isomorphism) purely in terms of their interactions between each other.

Lemma 1. Yoneda lemma*: For any functor $F : C \to \mathbf{Set}$, whose domain category C is "locally small" (meaning that the collection of morphisms between each pair of objects forms a set), any object c in C, there is a bijection*

$$Hom(C(c, -), F) \simeq Fc$$

that defines a natural transformation $\alpha : C(c, -) \Rightarrow F$ to the element $\alpha_c(1_c) \in Fc$. This correspondence is natural in both c and F.

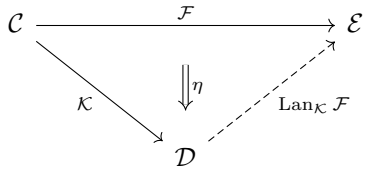
There is of course a dual form of the Yoneda Lemma in terms of the contravariant functor $C(-, c)$ as well using the natural transformation $C(-, c) \Rightarrow F$. A very useful way to interpret the Yoneda Lemma is through the notion of universal representability through a covariant or contravariant functor.

Definition 5. *A* **universal representation** *of an object $c \in C$ in a category C is defined as a contravariant functor F together with a functorial representation $C(-, c) \simeq F$ or by a covariant functor F together with a representation $C(c, -) \simeq F$. The collection of morphisms $C(-, c)$ into an object c is called the* **presheaf***, and from the Yoneda Lemma, forms a universal representation of the object.*

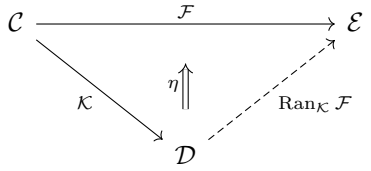
Kan Extensions

Often, in machine learning, we are given samples of a function defined on some subset $f : A \to B$ and we want to extend the function over a larger set $A \subset M$, but there is no obvious or canonical extension of functions. This ill-defined nature of machine learning has prompted a large variety of solutions, such as regularization or Occam's razor (prefer the simplest function). In contrast, if we are given a functor $F : A \to B$, and we want to extend the functor to a larger category M, there are only two canonical solutions that present themselves.

Definition 6. *A* **left Kan extension** *of a functor $\mathcal{F} : \mathcal{C} \to \mathcal{E}$ along another functor $\mathcal{K} : \mathcal{C} \to \mathcal{D}$, is a functor $Lan_{\mathcal{K}}\mathcal{F} : \mathcal{D} \to \mathcal{E}$ with a natural transformation $\eta : F \Rightarrow Lan_F \circ K$ such that for any other such pair $(G : \mathcal{D} \to \mathcal{E}, \gamma : F \Rightarrow GK)$, γ factors uniquely through η. In other words, there is a unique natural transformation $\alpha : Lan_F \Rightarrow G$.*



Definition 7. *A* **right Kan extension** *of a functor $\mathcal{F} : \mathcal{C} \to \mathcal{E}$ along another functor $\mathcal{K} : \mathcal{C} \to \mathcal{D}$, is a functor $\eta : Ran_F \circ K \to F$ with a natural transformation $\eta : Lan_F \circ K \Rightarrow \mathcal{F}$ such that for any other such pair $(G : \mathcal{D} \to \mathcal{E}, \gamma : GK \Rightarrow F)$, γ factors uniquely through η. In other words, there is a unique natural transformation $\alpha : G \Rightarrow Ran_F$.*



Extending functors is a central notion in category theory: Kan extension not only provides a canonical solution, but as MacLane (1971) famously remarked: "All concepts are Kan extensions". Remarkably, all the category-theoretic concepts in this section, from the Yoneda Lemma to adjoint functors can be in turn defined through the Kan extension!

Adjoint Functors

An important setting in many applications of AI and ML is where we have data or knowledge about a source domain (or category) and desire to make inferences about a target domain (category). This problem can be formalized using the concept of *adjoint functors*. Adjunctions are defined by an opposing pair of functors $F : C \leftrightarrow D : G$ that can be defined more precisely as follows.

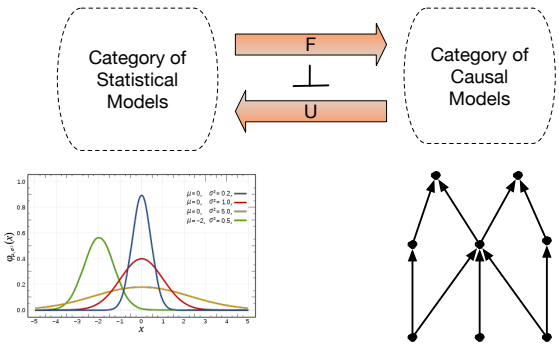


Figure 4: Adjoint functors between the category of statistical models causal models. Statistical models can be viewed as the result of applying a "forgetful" functor to a causal model that drops the directional structure in a causal model, whereas causal models can be viewed as "words" in a "free" algebra that results from the left adjoint functor to the forgetful functor.

Definition 8. *An* **adjunction** *consists of a pair of functors* $F : C \to D$ *and* $G : D \to C$*, where* F *is often referred to* left adjoint *and* G *is referred to as the* right adjoint*, that result in the following isomorphism relationship holding between their following sets of homomorphisms in categories* C *and* D*:*

$$D(Fc, d) \simeq C(c, Gd)$$

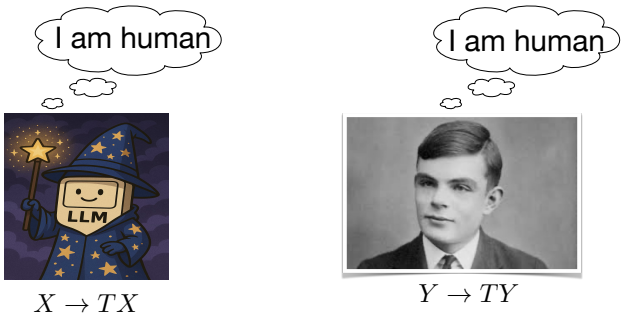
It is common to denote adjoint functors in this turnstile notation, indicating that $F : C \to D$ is left adjoint to $G : D \to C$, or more simply as $F \vdash G$.

$$\mathcal{D} \underset{F}{\overset{G}{\rightleftarrows}} \mathcal{C}.$$

The manifold learning method UMAP (Uniform Manifold Approximation and Projection) (McInnes, Healy, and Melville 2018) is based on constructing adjoint functors between the category of ultrametric spaces (these allow distances to be ∞) and the category of topological spaces. UMAP is currently one of the best and fastest methods for visualizing high-dimensional data in lower-dimensional spaces, outperforming Hinton's stochastic neighborhood embedding (t-SNE).

The most common type of adjoint functors is between categories $\mathcal{C}$ and $\mathcal{D}$ is where $\mathcal{F} : \mathcal{C} \to \mathcal{D}$ is a "free" functor, and the reverse $\mathcal{G} : \mathcal{D} \to \mathcal{F}$ is a "forgetful" functor. One interesting application of free-forgetful functors is to Pearl's Ladder of Causality (Pearl and Mackenzie 2018). Here, the bottommost "statistical" layer is adjoint to the middle "causal" layer.

Figure 4 illustrates the relationship between a category of statistical models and a category of causal models in terms of a pair of adjoint "forgetful-free" functors. A statistical model can be abstractly viewed in terms of its conditional independence properties. Similarly, in causal inference, $(x \perp y|z) \Rightarrow p(x, y, z) = p(x|z)p(y|z)$ denotes a statement about the probabilistic conditional independence of x and y given z. In causal inference, the goal is to recover a partial order defined as a directed acyclic graph (DAG) that ascribes causality among a set of random variables from a dataset specifying a sample of their joint distribution. It is well-known that without interventions, causality cannot be inferred uniquely, since because of Bayes rule, there is no purely observational approach to distinguish causal models such as $x \to y \to z$ from $z \to y \to x$.



$X \to TX$ $\quad$ $Y \to TY$

Figure 5: Turing's imitation game can be categorically formulated in terms of comparing two generative models defined as universal coalgebras (Jacobs 2016; Rutten 2000).

Categorical AGI: Universal Imitation Games

We have recently proposed how to model Artificial General Intelligence (AGI) using a category-theoretic framework called *Universal Imitation Games* (Mahadevan 2024). In Turing's imitation game framework, the identity of the two participants must be inferred from remote interactions (see Figure 5). We can model each participant as a generative model, defined as a coalgebra $X \to T(X)$, where X is the "state space" type and T is a functor that specifies the model. Jacobs (2016) contains an excellent overview of how to specify a broad range of generative models as coalgebras, from deterministic models such as finite state machines and grammars, deep learning models such as LLMs (Mahadevan 2025d), as well as stochastic models such as Markov chains or partially observable MDPs (Mahadevan 2025f) We can formalize Turing's imitation game as checking if two coalgebras are isomorphic, which is well-studied in the literature on universal coalgebras (Rutten 2000). In Turing's original formulation, the participants were assumed to be *static*. When participants adapt during interactions, we can define two special cases of UIGs: *dynamic UIGs* and *evolutionary UIGs*. In dynamic UIGs, the participants are adapting from their interactions, which captures the process of training LLMs from human feedback. In evolutionary UIGs, the participants evolve, much like biological systems, based on their competition with each other. As AGI systems continue to become widely deployed and interact with humans and other AGI systems, we are likely to see evolutionary adaptation in these systems as well.

Acknowledgments

This research is funded by Adobe Corporation.

References

Bell, J. L. 1988. *Toposes and Local Set Theories*. Dover.

Bengio, Y. 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1): 1–127.

Bertsekas, D. 2019. *Reinforcement Learning and Optimal Control*. Athena Scientific.

Bradley, T.-D.; Terilla, J.; and Vlassopoulos, Y. 2022. An Enriched Category Theory of Language: From Syntax to Semantics. *La Matematica*.

Carlsson, G. E.; and Mémoli, F. 2013. Classifying Clustering Schemes. *Found. Comput. Math.*, 13(2): 221–252.

Cho, K.; and Jacobs, B. 2019. Disintegration and Bayesian inversion via string diagrams. *Mathematical Structures in Computer Science*, 29(7): 938–971.

DeepSeek-AI. 2024. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism. *CoRR*, abs/2401.02954.

Dziri, N.; Lu, X.; Sclar, M.; Li, X. L.; Jiang, L.; Lin, B. Y.; West, P.; Bhagavatula, C.; Bras, R. L.; Hwang, J. D.; Sanyal, S.; Welleck, S.; Ren, X.; Ettinger, A.; Harchaoui, Z.; and Choi, Y. 2023. Faith and Fate: Limits of Transformers on Compositionality. arXiv:2305.18654.

Fong, B. 2012. *Causal Theories: A Categorical Perspective on Bayesian Networks*. Master's thesis, Oxford University.

Fong, B.; and Spivak, D. I. 2018. *Seven Sketches in Compositionality: An Invitation to Applied Category Theory*. Cambridge University Press.

Fong, B.; Spivak, D. I.; and Tuyéras, R. 2019. Backprop as Functor: A compositional perspective on supervised learning. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, 1–13. IEEE.

Forré, P.; and Mooij, J. M. 2017. Markov Properties for Graphical Models with Cycles and Latent Variables. arXiv:1710.08775.

Fritz, T. 2020. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370: 107239.

Fritz, T.; and Klingler, A. 2023. The d-Separation Criterion in Categorical Probability. *Journal of Machine Learning Research*, 24(46): 1–49.

Galles, D.; and Pearl, J. 1988. An Axiomatic theory of counterfactuals. *Foundations of Science*, 3: 151–182.

Heunen, C.; and Vicary, J. 2019. *Categories for Quantum Theory: An Introduction*. Oxford University Press.

Imbens, G. W.; and Rubin, D. B. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. USA: Cambridge University Press. ISBN 0521885884.

Jacobs, B. 2016. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press. ISBN 9781316823187.

Jacobs, B.; Kissinger, A.; and Zanasi, F. 2018. Causal Inference by String Diagram Surgery.

Kleinberg, J. 2002. An Impossibility Theorem for Clustering. In Becker, S.; Thrun, S.; and Obermayer, K., eds., *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

Lurie, J. 2009. *Higher Topos Theory*. Annals of mathematics studies. Princeton, NJ: Princeton University Press.

MacLane, S. 1971. *Categories for the Working Mathematician*. New York: Springer-Verlag. Graduate Texts in Mathematics, Vol. 5.

Mahadevan, S. 2021. Universal Decision Models. *CoRR*, abs/2110.15431.

Mahadevan, S. 2023. Universal Causality. *Entropy*, 25(4): 574.

Mahadevan, S. 2024. Universal Imitation Games. arXiv:2405.01540.

Mahadevan, S. 2025a. Decentralized Causal Discovery using Judo Calculus. arXiv:2510.23942.

Mahadevan, S. 2025b. Higher Algebraic K-Theory of Causality. *Entropy*, 27(5).

Mahadevan, S. 2025c. Intuitionistic j-Do-Calculus in Topos Causal Models. arXiv:2510.17944.

Mahadevan, S. 2025d. Topos Theory for Generative AI and LLMs. arXiv:2508.08293.

Mahadevan, S. 2025e. Universal Causal Inference in a Topos. In *Advances in Neural Information Processing Systems*, volume 39.

Mahadevan, S. 2025f. Universal Reinforcement Learning in Coalgebras: Asynchronous Stochastic Computation via Conduction. arXiv:2508.15128.

McInnes, L.; Healy, J.; and Melville, J. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.

Parascandolo, G.; Rojas-Carulla, M.; Kilbertus, N.; and Schölkopf, B. 2017. Learning Independent Causal Mechanisms. *CoRR*, abs/1712.00961.

Pearl, J. 2009. *Causality: Models, Reasoning and Inference*. USA: Cambridge University Press, 2nd edition. ISBN 052189560X.

Pearl, J.; and Mackenzie, D. 2018. *The Book of Why: The New Science of Cause and Effect*. USA: Basic Books, Inc., 1st edition. ISBN 046509760X.

Rezaei-Shoshtari, S.; Zhao, R.; Panangaden, P.; Meger, D.; and Precup, D. 2022. Continuous MDP Homomorphisms and Homomorphic Policy Gradient. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Richter, B. 2020. *From Categories to Homotopy Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press. ISBN 9781108479622.

Riehl, E. 2017. *Category Theory in Context*. Aurora: Dover Modern Math Originals. Dover Publications. ISBN 9780486820804.

Rutten, J. J. M. M. 2000. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1): 3–80.

Selinger, P. 2010. A Survey of Graphical Languages for Monoidal Categories. In *New Structures for Physics*, 289–355. Springer Berlin Heidelberg.

Spirtes, P.; Glymour, C.; and Scheines, R. 2000. *Causation, Prediction, and Search, Second Edition*. Adaptive computation and machine learning. MIT Press. ISBN 978-0-262-19440-2.

Studeny, M. 2010. *Probabilistic Conditional Independence Structures*. Information Science and Statistics. Springer London. ISBN 9781849969482.

Vafa, K.; Chang, P. G.; Rambachan, A.; and Mullainathan, S. 2025. What Has a Foundation Model Found? Using Inductive Bias to Probe for World Models. arXiv:2507.06952.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.