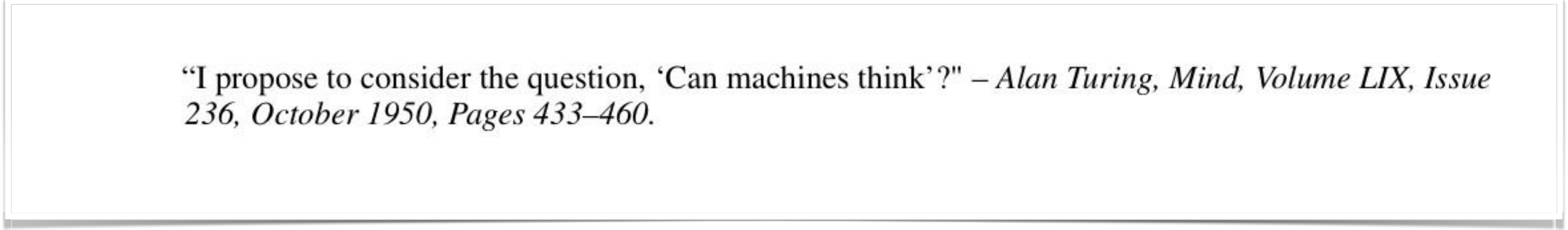
Universal Imitation Games: Generative AI beyond deep learning

Sridhar Mahadevan
Adobe Research and the University of Massachusetts

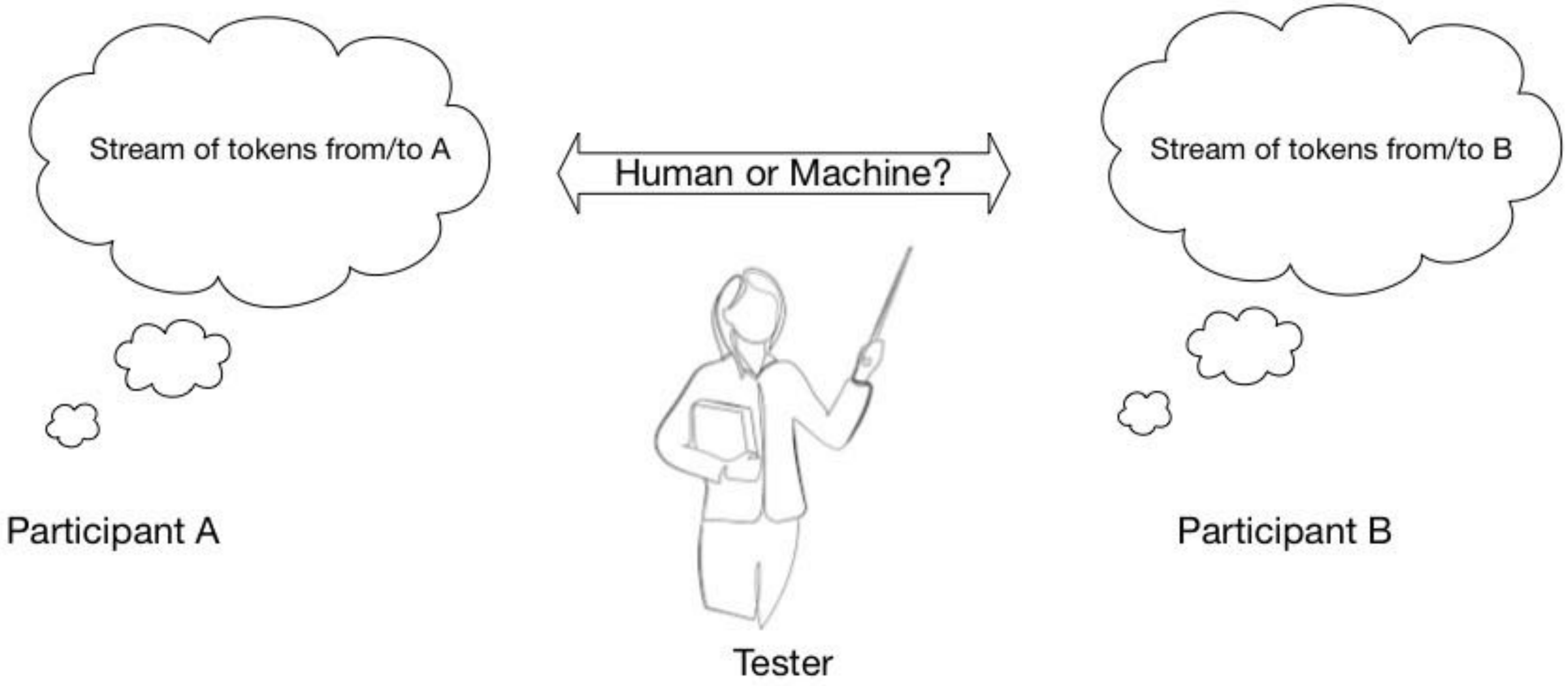
Overview of Talks

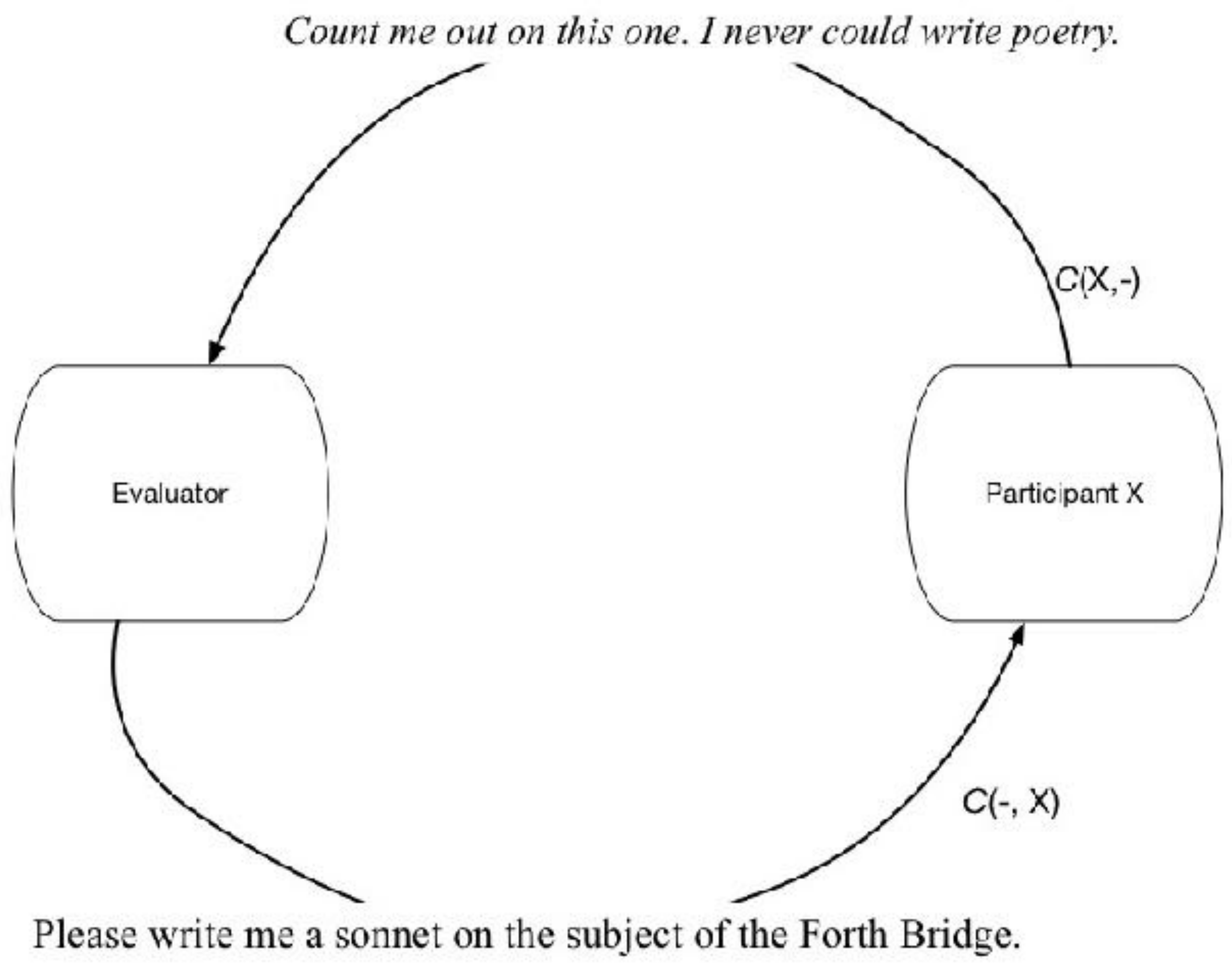
- Today’s talk: GAIA: A Generative AI Architecture beyond deep learning
- Tomorrow’s talk: Generative AI using Universal Coalgebras
- Wednesday’s talk: The (co)End of Generative AI Models

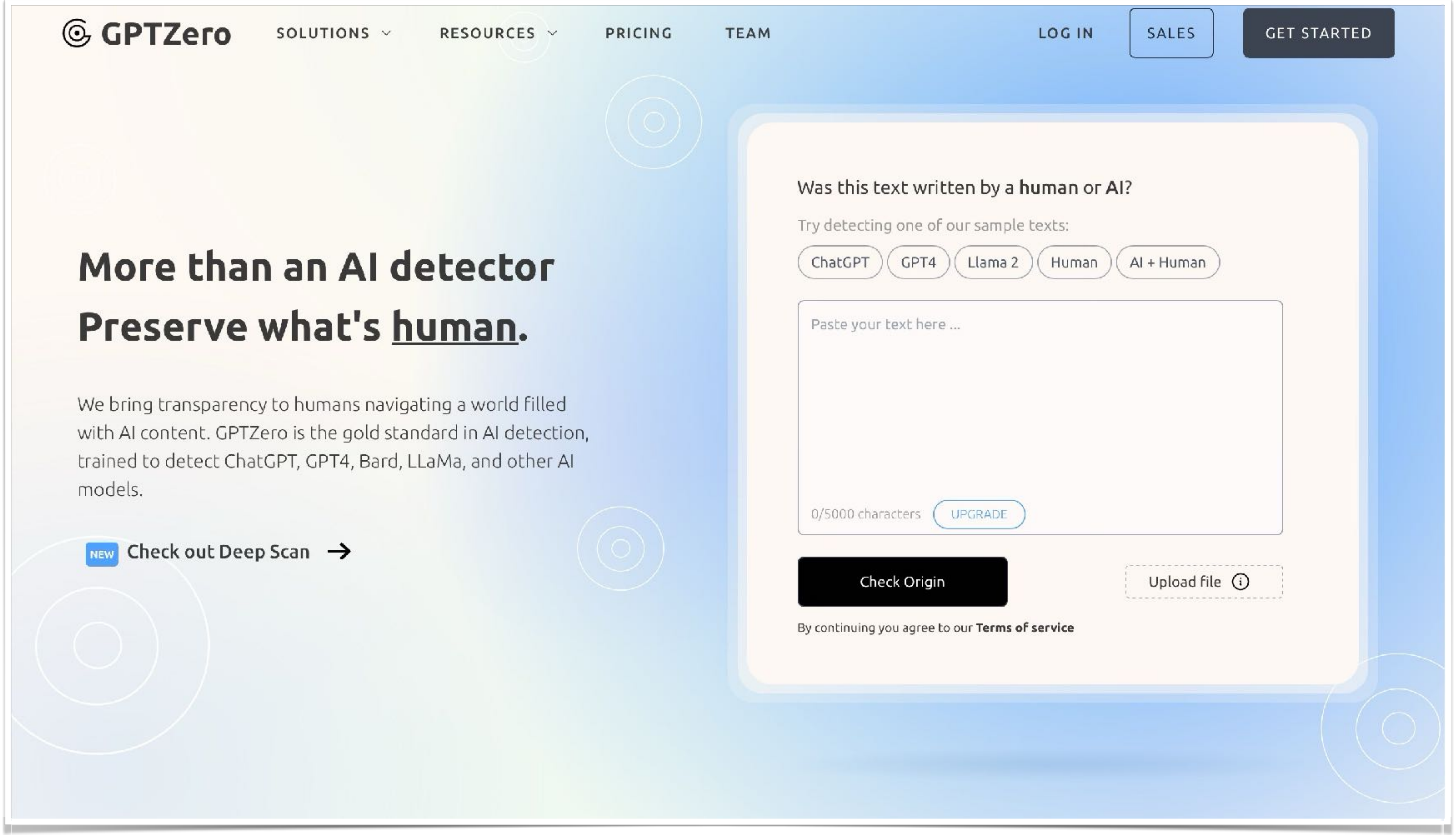


“I propose to consider the question, ‘Can machines think’?" – *Alan Turing, Mind, Volume LIX, Issue 236, October 1950, Pages 433–460.*

Imitation Games


Stream of tokens from/to A
Human or Machine?
Stream of tokens from/to B
Participant A
Participant B
Tester


Count me out on this one. I never could write poetry.
C(X,-)
Evaluator
Participant X
C(-, X)
Please write me a sonnet on the subject of the Forth Bridge.



More than an AI detector
Preserve what's human.

We bring transparency to humans navigating a world filled with AI content. GPTZero is the gold standard in AI detection, trained to detect ChatGPT, GPT4, Bard, LLaMa, and other AI models.

Was this text written by a **human** or **AI**?

Try detecting one of our sample texts:

By continuing you agree to our **Terms of service**

Paper online at my

UMass home page

Forthcoming book!

UNIVERSAL IMITATION GAMES*

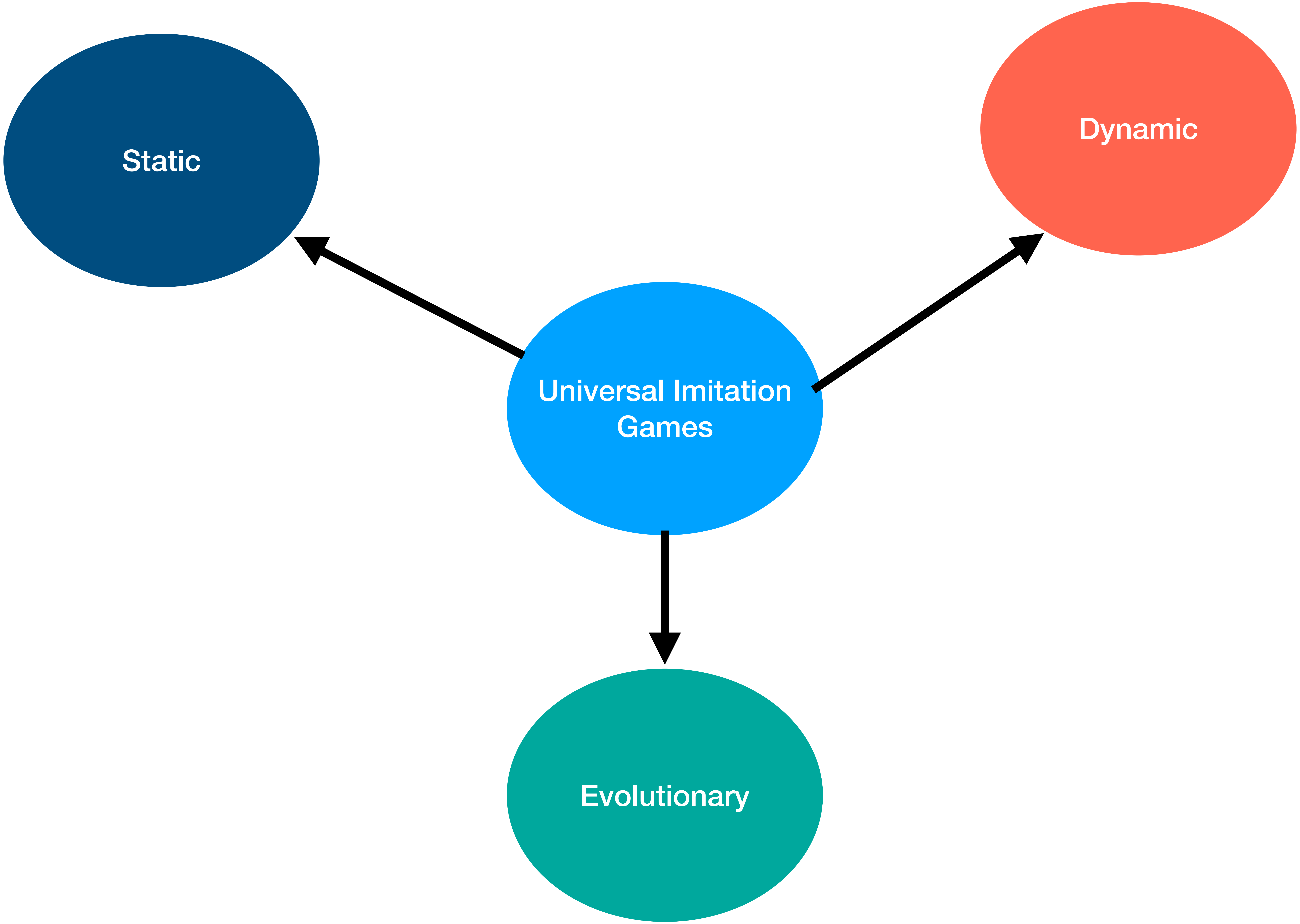
A PREPRINT

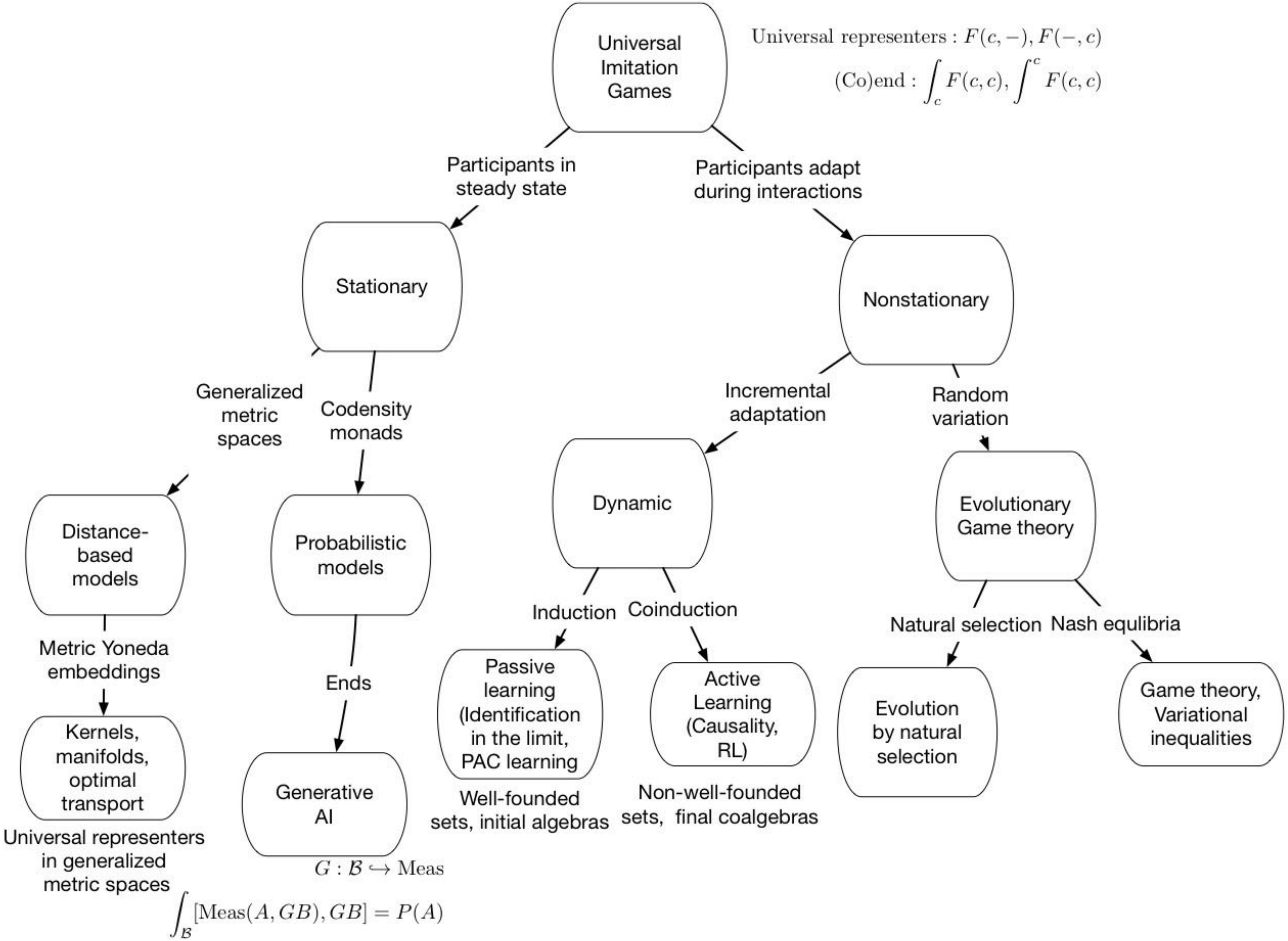
Sridhar Mahadevan
Adobe Research and University of Massachusetts, Amherst
smahadev@adobe.com, mahadeva@umass.edu

February 16, 2024

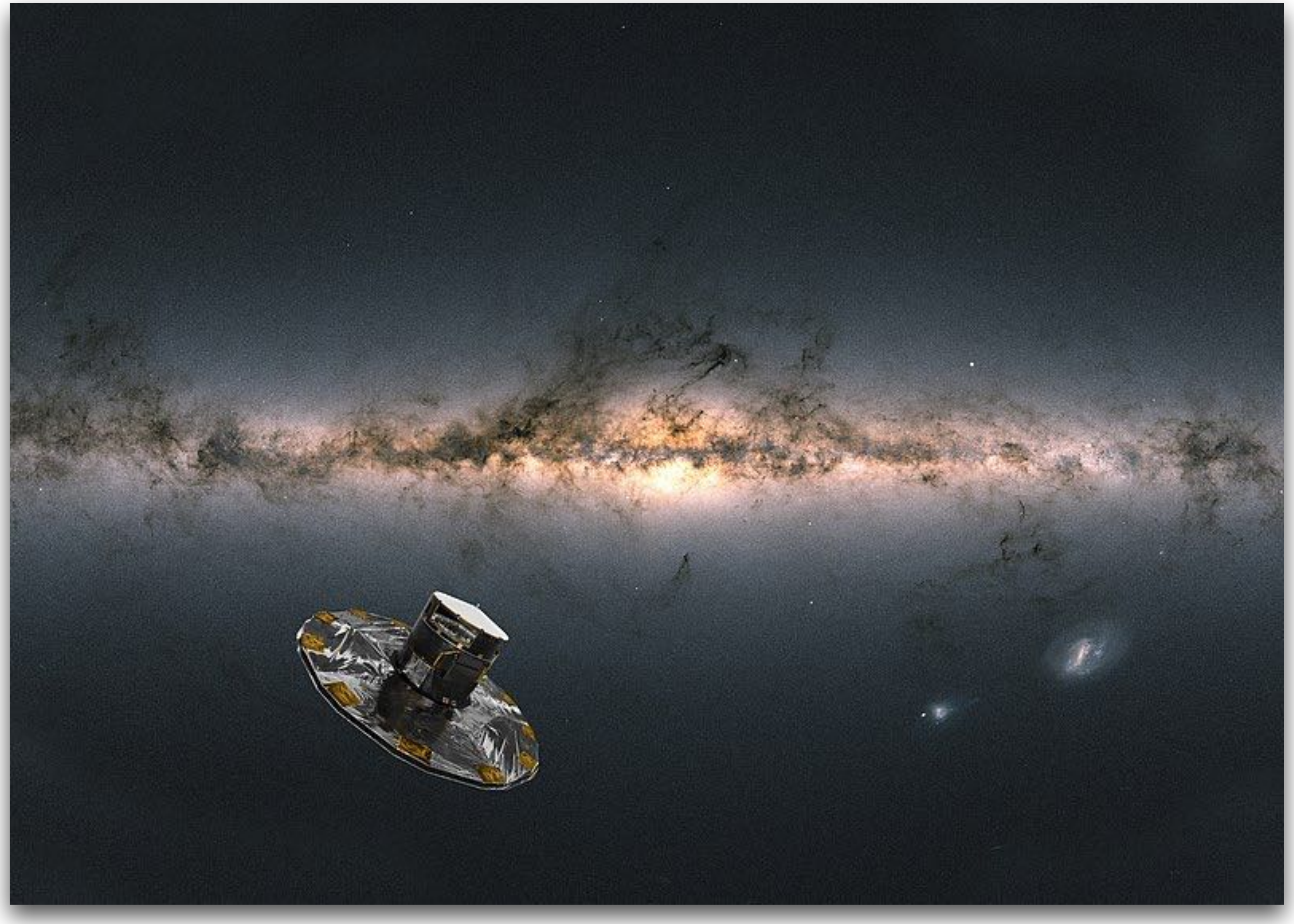
ABSTRACT

In 1950, Alan Turing proposed a framework called an *imitation game* in which the participants are to be classified `Human or Machine` solely from natural language interactions. Using mathematics largely developed since Turing – category theory – we investigate a broader class of *universal imitation games* (UIGs). Choosing a category means defining a collection of objects and a collection of composable arrows between each pair of objects that represent "measurement probes" for solving UIGs. The theoretical foundation of our paper rests on two celebrated results by Yoneda. The first, called the Yoneda Lemma, discovered in 1954 – the year of Turing's death – shows that objects in categories can be identified up to isomorphism solely with measurement probes defined by composable arrows. Yoneda embeddings are universal representers of objects in categories. A simple yet general solution to the static UIG problem, where the participants are not changing during the interactions, is to determine if the Yoneda embeddings are (weakly) isomorphic. A *universal property* in category theory is defined by an *initial* or *final* object. A second foundational result of Yoneda from 1960 defines initial objects called *coends* and final objects called *ends*, which yields a categorical "integral calculus" that unifies probabilistic generative models, distance-based kernel, metric and optimal transport models, as well as topological manifold representations. When participants adapt during interactions, we study two special cases: in *dynamic UIGs*, "learners" imitate "teachers". We contrast the initial object framework of *passive learning from observation* over well-founded sets using inductive inference – extensively studied by Gold, Solomonoff, Valiant, and Vapnik – with the final object framework of *coinductive inference* over non-well-founded sets and universal coalgebras, which formalizes learning from *active experimentation* using causal inference or reinforcement


Static
Dynamic
Universal Imitation Games
Evolutionary

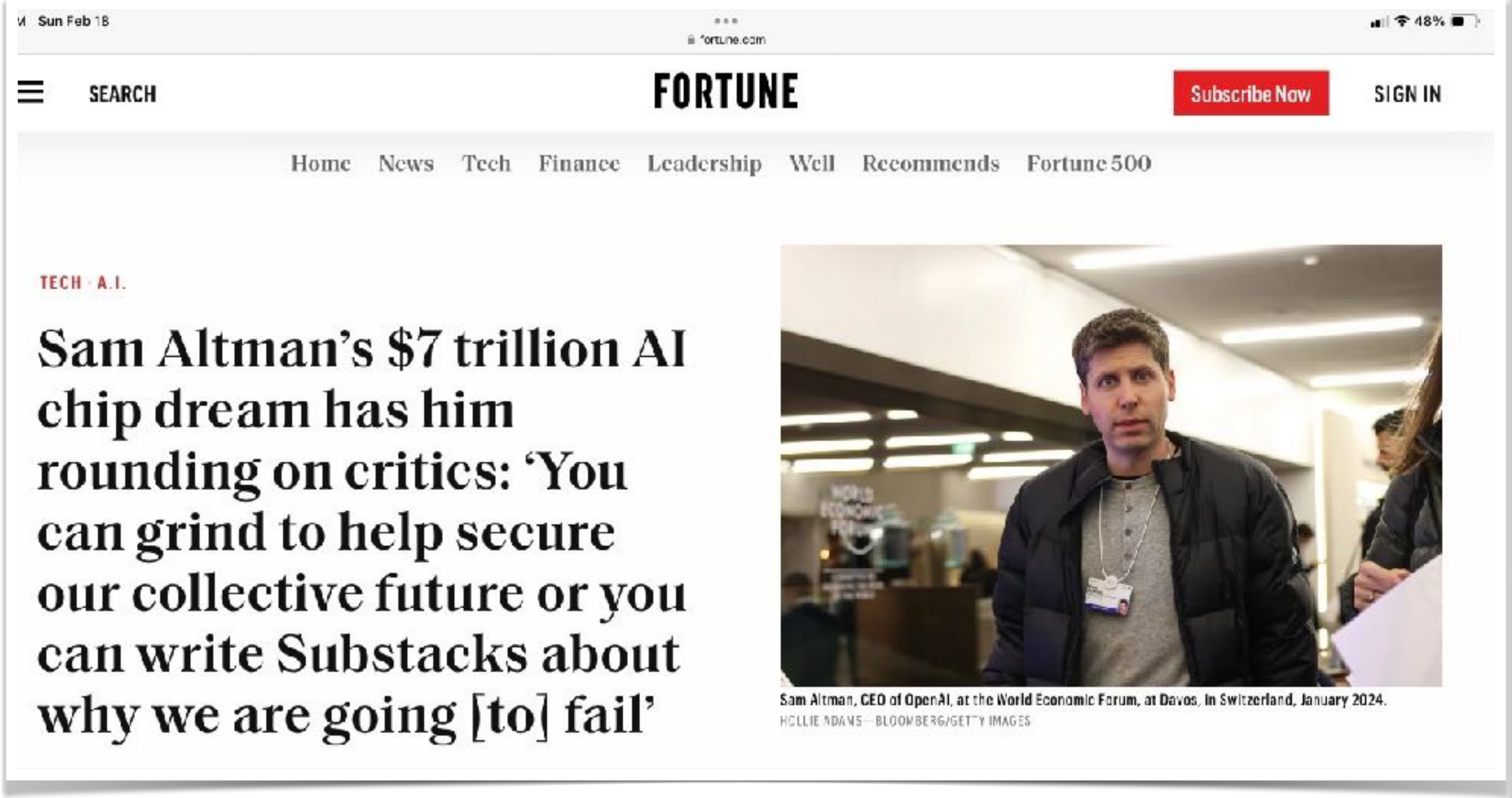

Universal Imitation Games
Universal representers : $F(c,-), F(-,c)$
(Co)end : $\int_c F(c,c), \int^c F(c,c)$
Participants in steady state
Participants adapt during interactions
Stationary
Nonstationary
Generalized metric spaces
Codensity monads
Incremental adaptation
Random variation
Distance-based models
Probabilistic models
Dynamic
Evolutionary Game theory
Metric Yoneda embeddings
Ends
Induction
Coinduction
Natural selection
Nash equlibria
Kernels, manifolds, optimal transport
Generative AI
Passive learning (Identification in the limit, PAC learning
Active Learning (Causality, RL)
Evolution by natural selection
Game theory, Variational inequalities
Universal representers in generalized metric spaces
$G : \mathcal{B} \hookrightarrow \text{Meas}$
$\int_{\mathcal{B}} [\text{Meas}(A, GB), GB] = P(A)$
Well-founded sets, initial algebras
Non-well-founded sets, final coalgebras

GAIA: Generative AI Architecture



Beyond Deep Learning!

Generative AI faces energy crisis

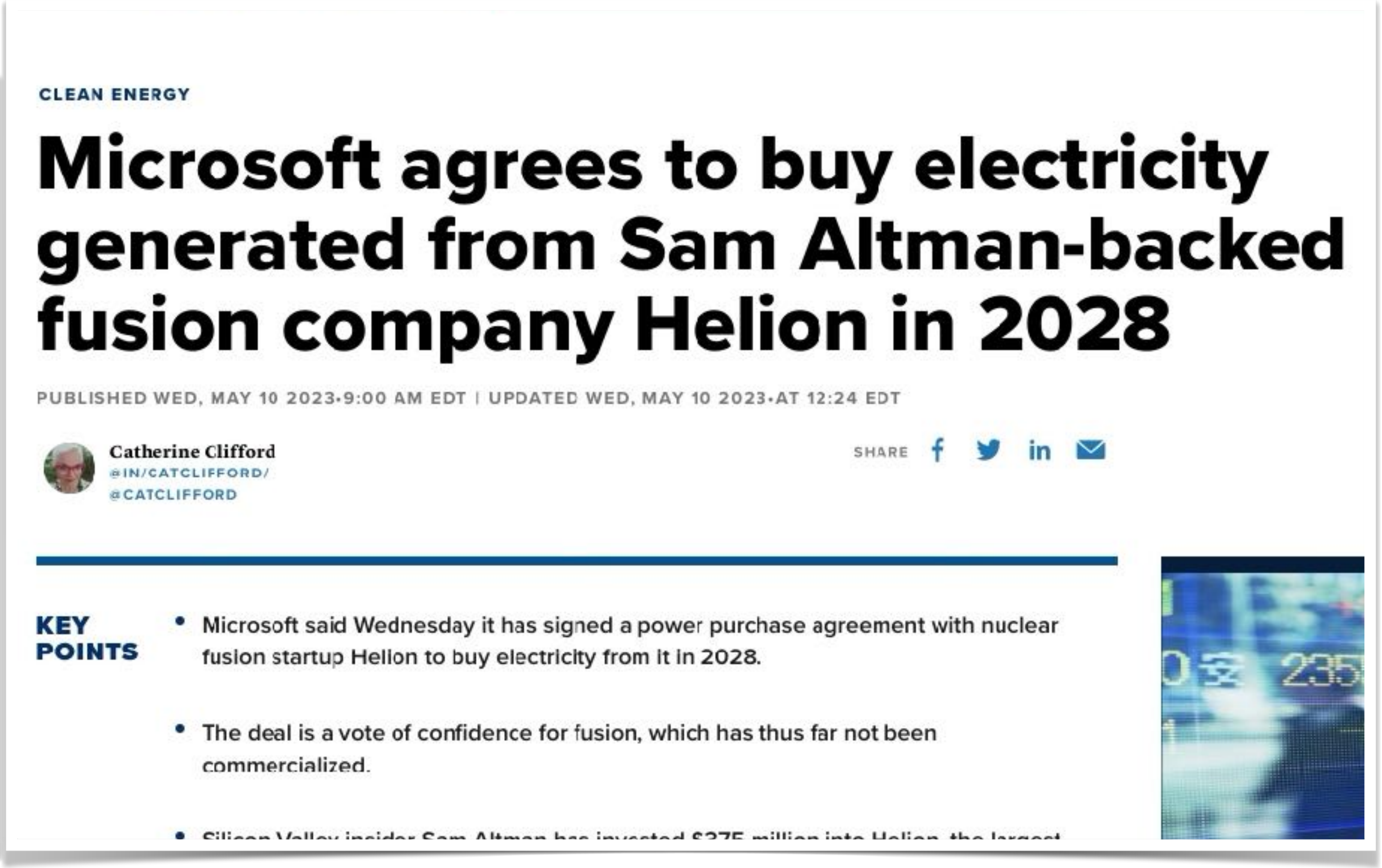


TECH · A.I.

Sam Altman's $7 trillion AI chip dream has him rounding on critics: 'You can grind to help secure our collective future or you can write Substacks about why we are going [to] fail'

Sam Altman, CEO of OpenAI, at the World Economic Forum, at Davos, in Switzerland, January 2024.

HOLLIE ADAMS—BLOOMBERG/GETTY IMAGES



CLEAN ENERGY

Microsoft agrees to buy electricity generated from Sam Altman-backed fusion company Helion in 2028

PUBLISHED WED, MAY 10 2023·9:00 AM EDT | UPDATED WED, MAY 10 2023·AT 12:24 EDT

Catherine Clifford
@IN/CATCLIFFORD/
@CATCLIFFORD

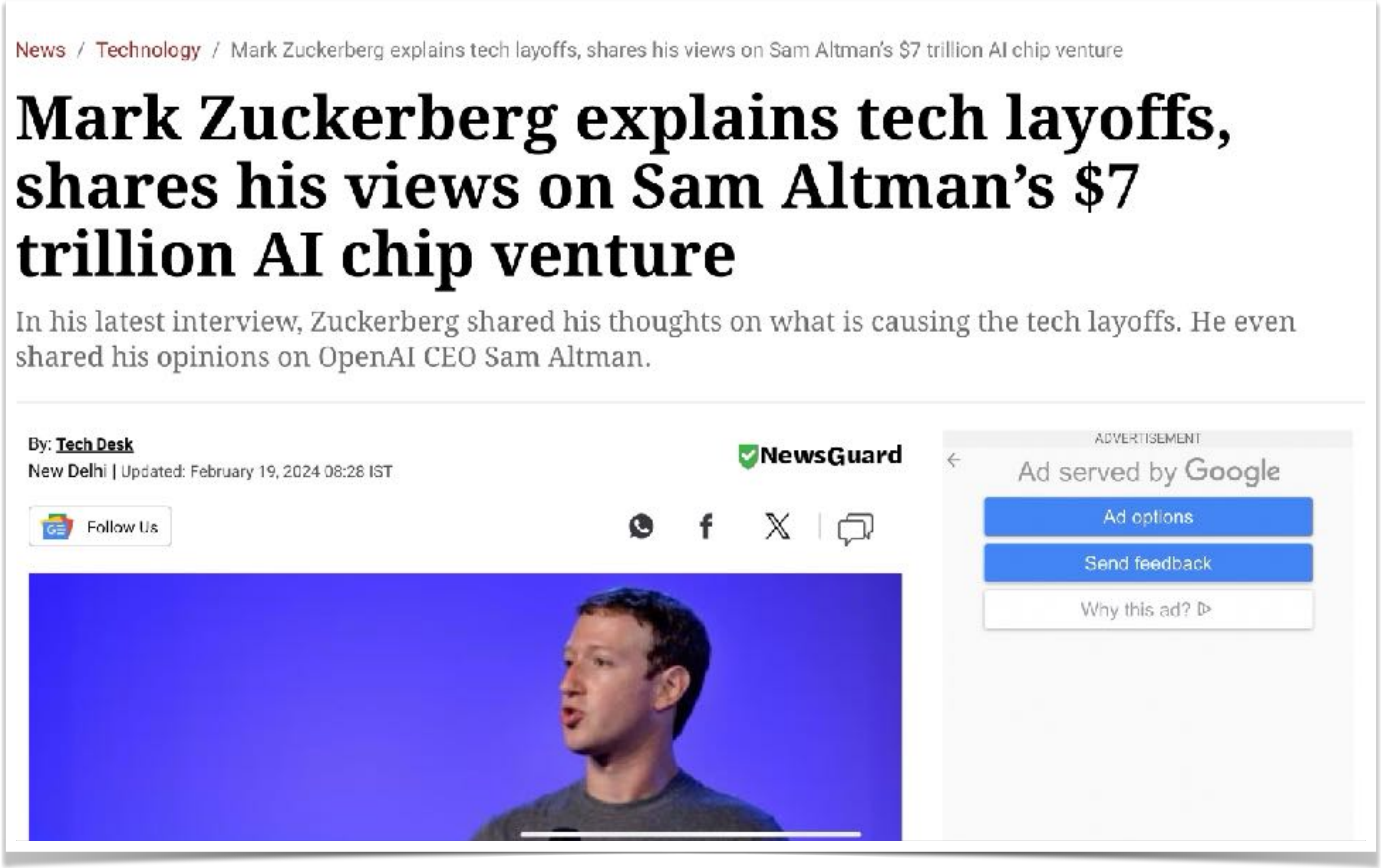
KEY POINTS

- Microsoft said Wednesday it has signed a power purchase agreement with nuclear fusion startup Helion to buy electricity from it in 2028.
- The deal is a vote of confidence for fusion, which has thus far not been commercialized.
- Silicon Valley insider Sam Altman has invested $375 million into Helion, the largest

Thursday, Feb 22, 2024

EPAPER | TODAY'S PAPER

The Indian EXPRESS
JOURNALISM OF COURAGE



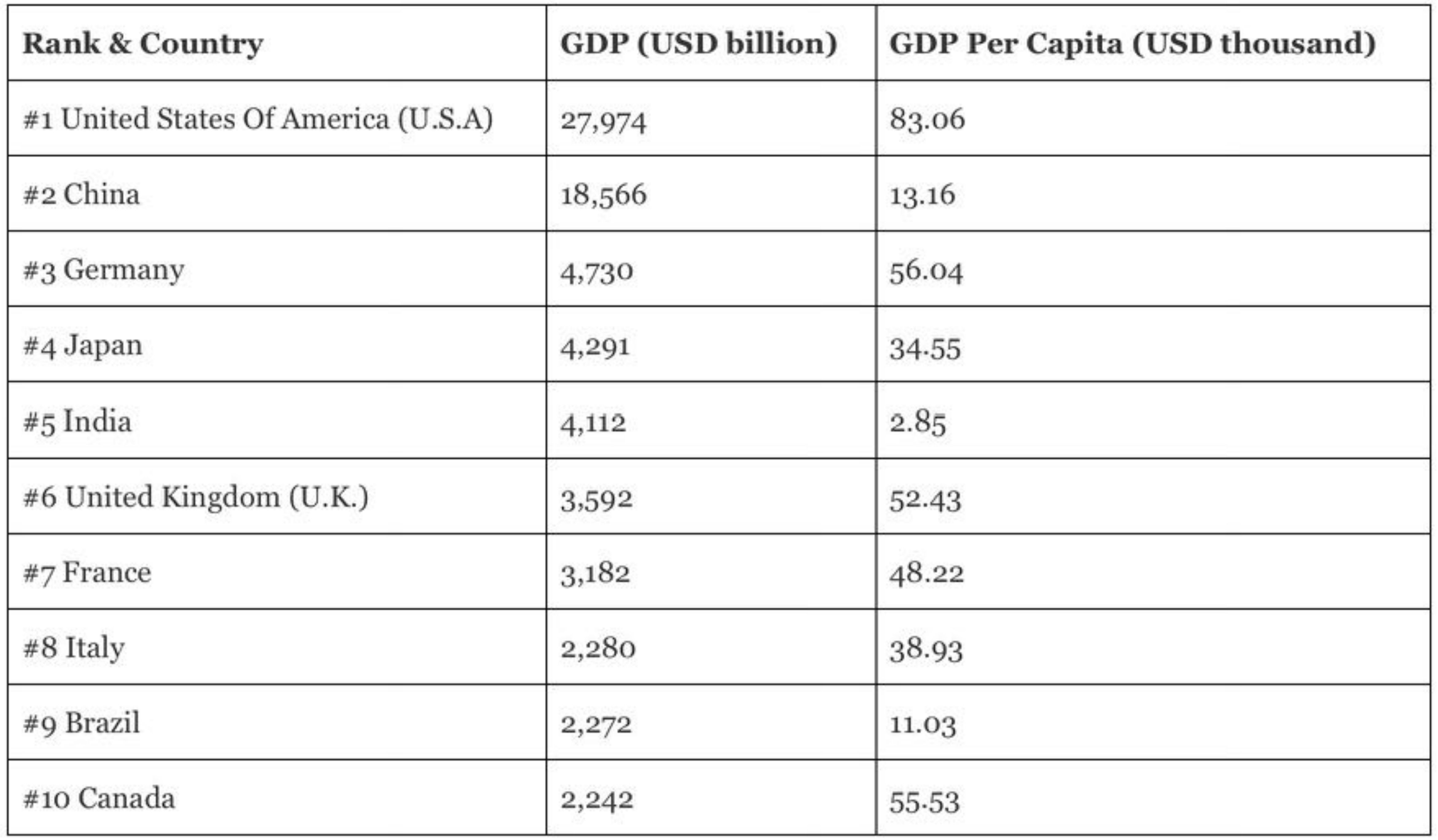
News / Technology / Mark Zuckerberg explains tech layoffs, shares his views on Sam Altman's $7 trillion AI chip venture

Mark Zuckerberg explains tech layoffs, shares his views on Sam Altman's $7 trillion AI chip venture

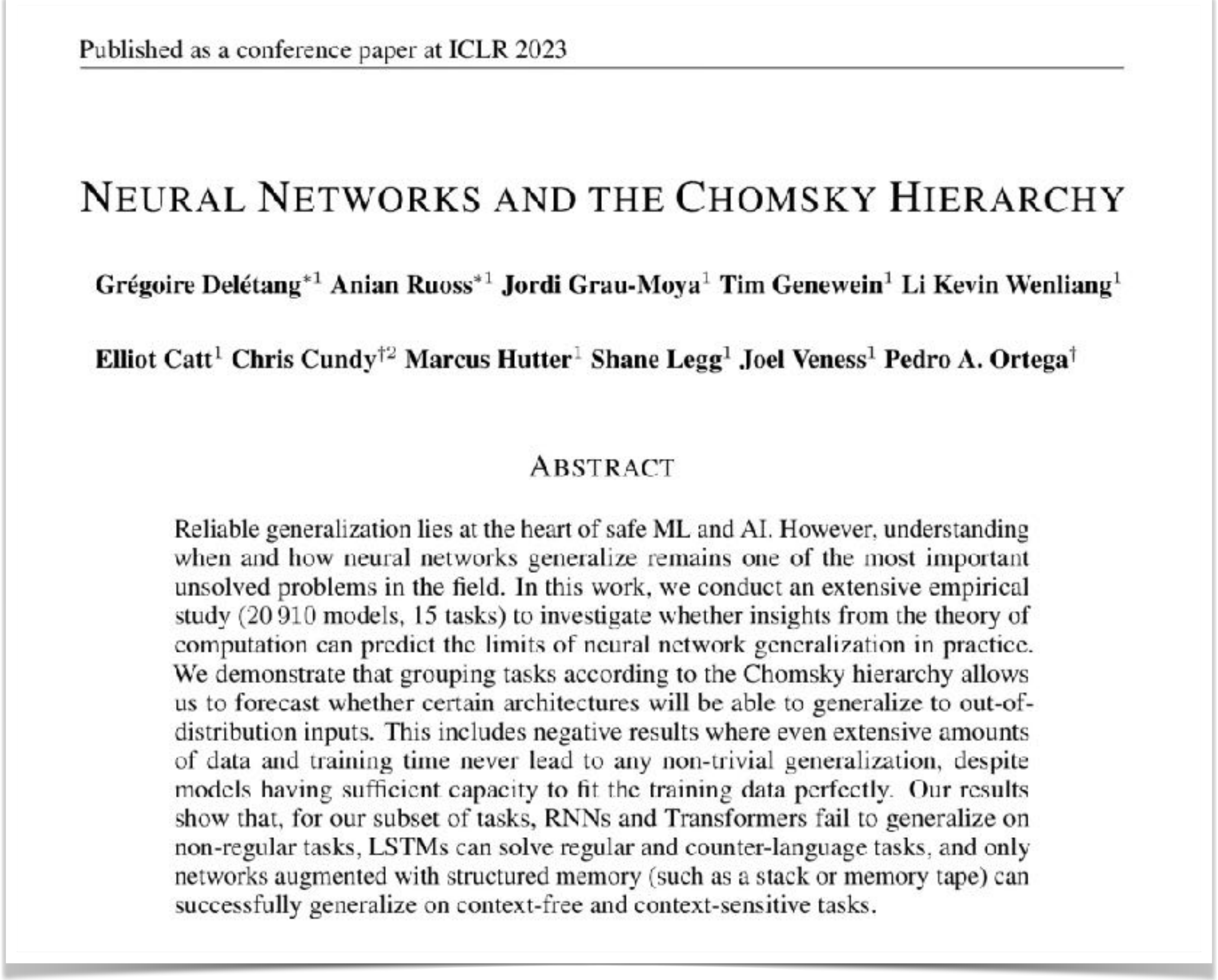
In his latest interview, Zuckerberg shared his thoughts on what is causing the tech layoffs. He even shared his opinions on OpenAI CEO Sam Altman.

By: Tech Desk
New Delhi | Updated: February 19, 2024 08:28 IST

NewsGuard



Rank & Country	GDP (USD billion)	GDP Per Capita (USD thousand)
#1 United States Of America (U.S.A)	27,974	83.06
#2 China	18,566	13.16
#3 Germany	4,730	56.04
#4 Japan	4,291	34.55
#5 India	4,112	2.85
#6 United Kingdom (U.K.)	3,592	52.43
#7 France	3,182	48.22
#8 Italy	2,280	38.93
#9 Brazil	2,272	11.03
#10 Canada	2,242	55.53



Published as a conference paper at ICLR 2023

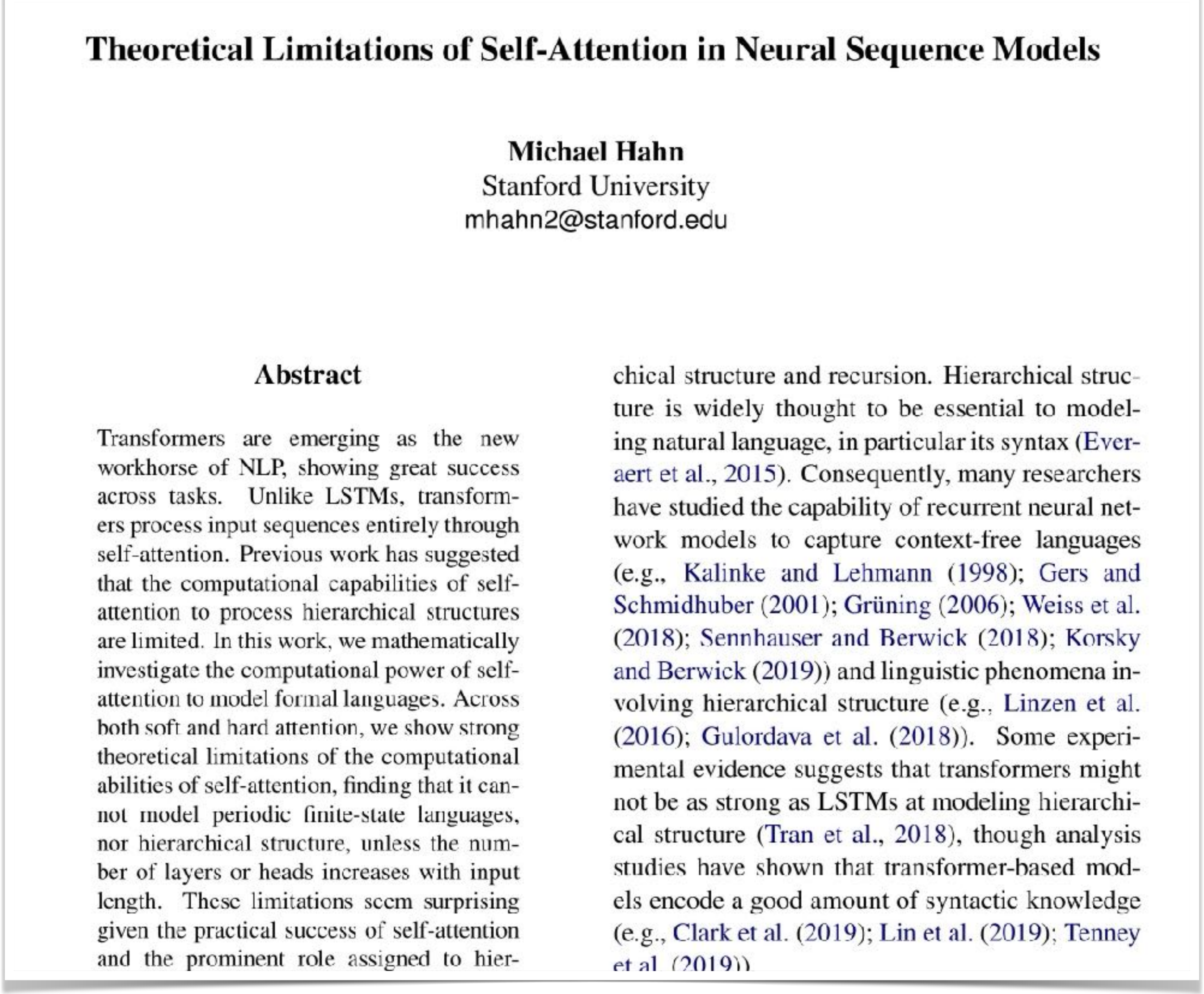
NEURAL NETWORKS AND THE CHOMSKY HIERARCHY

Grégoire Delétang[*1] Anian Ruoss[*1] Jordi Grau-Moya[1] Tim Genewein[1] Li Kevin Wenliang[1]

Elliot Catt[1] Chris Cundy[†2] Marcus Hutter[1] Shane Legg[1] Joel Veness[1] Pedro A. Ortega[†]

ABSTRACT

Reliable generalization lies at the heart of safe ML and AI. However, understanding when and how neural networks generalize remains one of the most important unsolved problems in the field. In this work, we conduct an extensive empirical study (20 910 models, 15 tasks) to investigate whether insights from the theory of computation can predict the limits of neural network generalization in practice. We demonstrate that grouping tasks according to the Chomsky hierarchy allows us to forecast whether certain architectures will be able to generalize to out-of-distribution inputs. This includes negative results where even extensive amounts of data and training time never lead to any non-trivial generalization, despite models having sufficient capacity to fit the training data perfectly. Our results show that, for our subset of tasks, RNNs and Transformers fail to generalize on non-regular tasks, LSTMs can solve regular and counter-language tasks, and only networks augmented with structured memory (such as a stack or memory tape) can successfully generalize on context-free and context-sensitive tasks.



Theoretical Limitations of Self-Attention in Neural Sequence Models

Michael Hahn
Stanford University
mhahn2@stanford.edu

Abstract

Transformers are emerging as the new workhorse of NLP, showing great success across tasks. Unlike LSTMs, transformers process input sequences entirely through self-attention. Previous work has suggested that the computational capabilities of self-attention to process hierarchical structures are limited. In this work, we mathematically investigate the computational power of self-attention to model formal languages. Across both soft and hard attention, we show strong theoretical limitations of the computational abilities of self-attention, finding that it cannot model periodic finite-state languages, nor hierarchical structure, unless the number of layers or heads increases with input length. These limitations seem surprising given the practical success of self-attention and the prominent role assigned to hierarchical structure and recursion. Hierarchical structure is widely thought to be essential to modeling natural language, in particular its syntax (Everaert et al., 2015). Consequently, many researchers have studied the capability of recurrent neural network models to capture context-free languages (e.g., Kalinke and Lehmann (1998); Gers and Schmidhuber (2001); Grüning (2006); Weiss et al. (2018); Sennhauser and Berwick (2018); Korsky and Berwick (2019)) and linguistic phenomena involving hierarchical structure (e.g., Linzen et al. (2016); Gulordava et al. (2018)). Some experimental evidence suggests that transformers might not be as strong as LSTMs at modeling hierarchical structure (Tran et al., 2018), though analysis studies have shown that transformer-based models encode a good amount of syntactic knowledge (e.g., Clark et al. (2019); Lin et al. (2019); Tenney et al. (2019))

Transformers cannot solve simple problems:

parity, integer modulo arithmetic, balancing arithmetic expressions

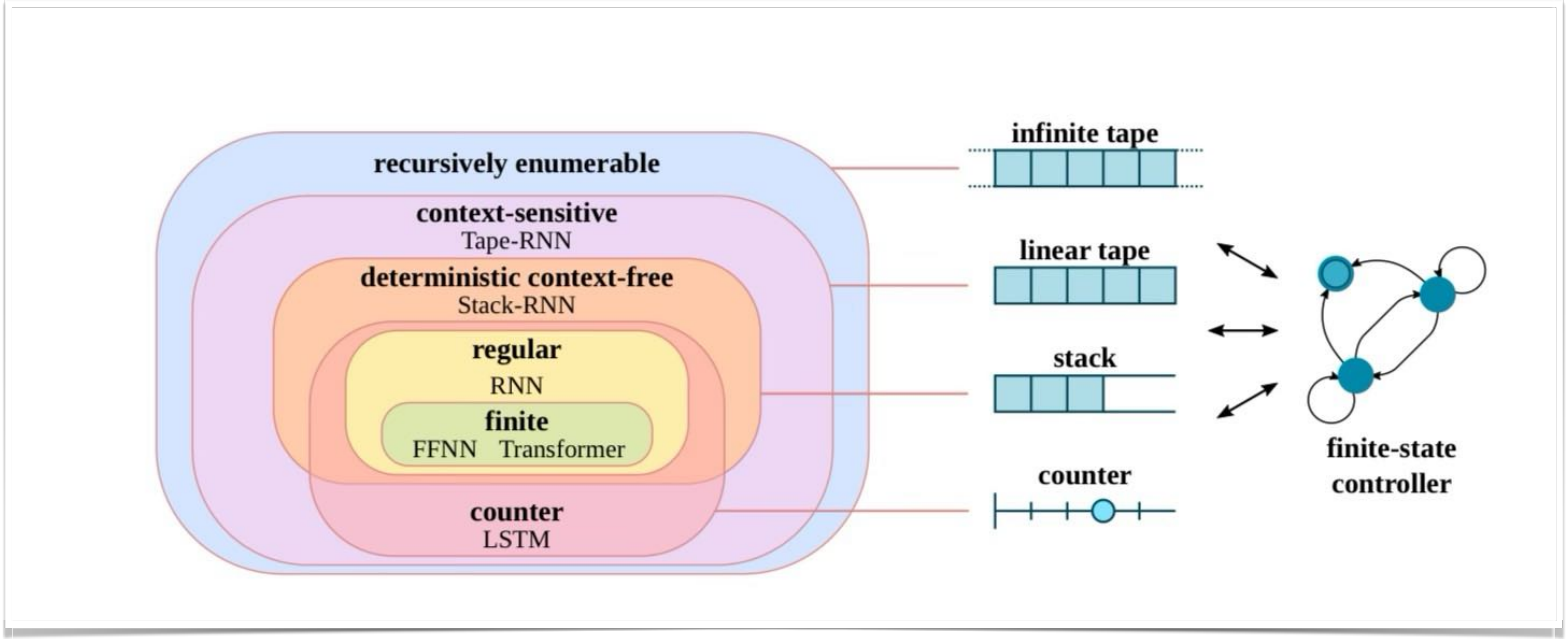
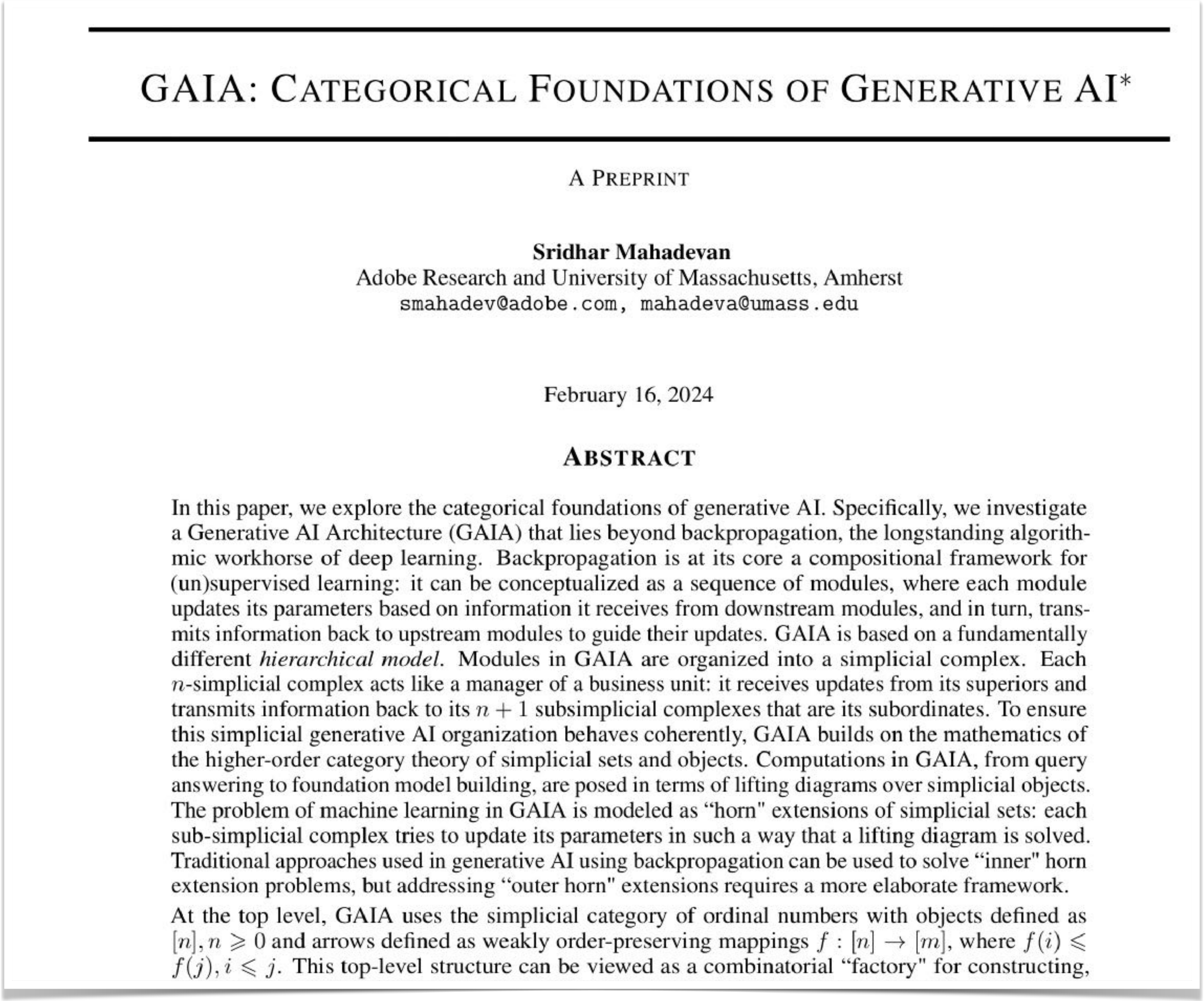


Figure source: Neural Networks and the Chomsky Hierarchy, ICLR 2023

Paper online at my

UMass home page

Forthcoming book!



GAIA: Categorical Foundations of Generative AI*

A Preprint

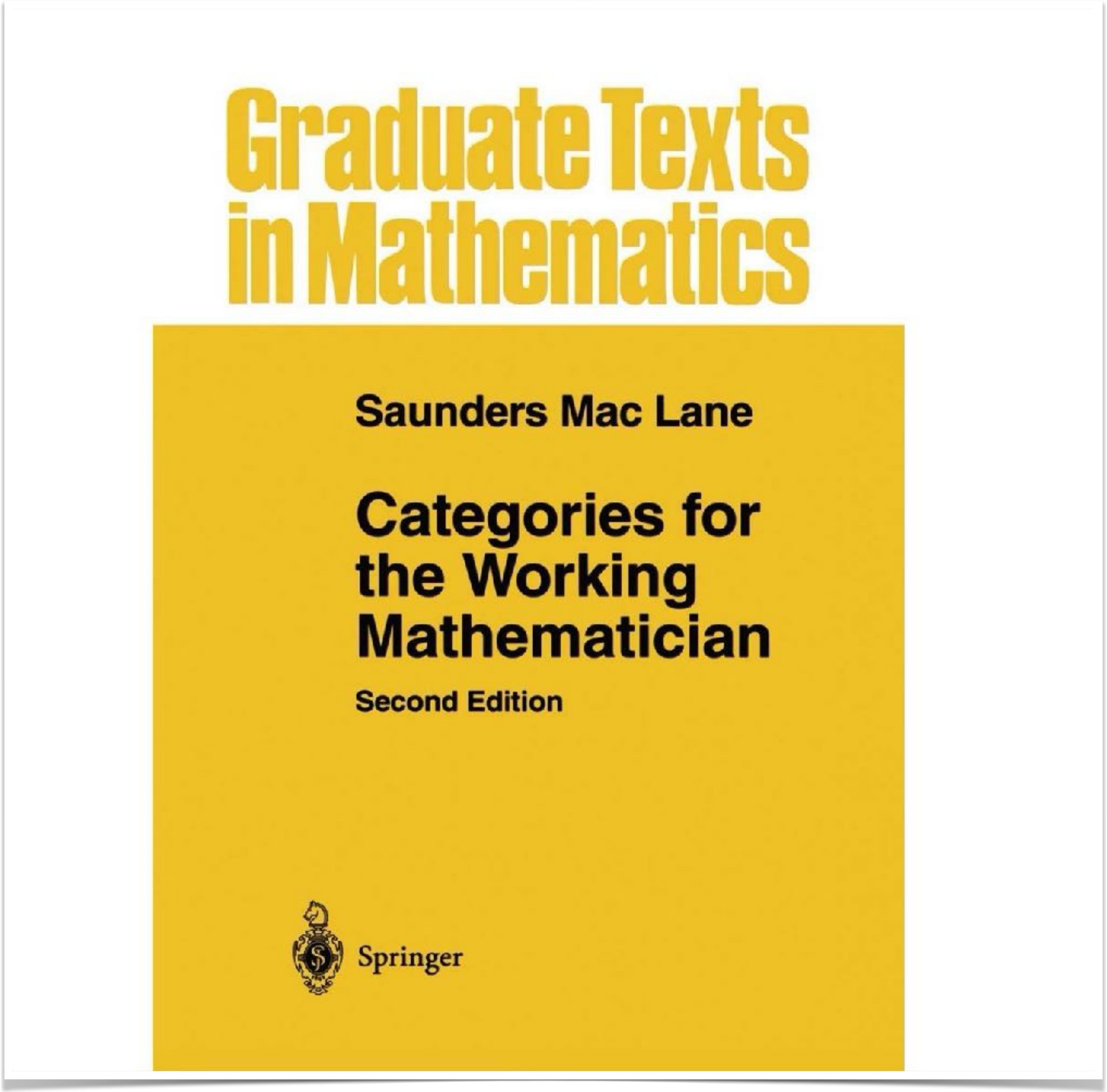
Sridhar Mahadevan
Adobe Research and University of Massachusetts, Amherst
smahadev@adobe.com, mahadeva@umass.edu

February 16, 2024

Abstract

In this paper, we explore the categorical foundations of generative AI. Specifically, we investigate a Generative AI Architecture (GAIA) that lies beyond backpropagation, the longstanding algorithmic workhorse of deep learning. Backpropagation is at its core a compositional framework for (un)supervised learning: it can be conceptualized as a sequence of modules, where each module updates its parameters based on information it receives from downstream modules, and in turn, transmits information back to upstream modules to guide their updates. GAIA is based on a fundamentally different *hierarchical model*. Modules in GAIA are organized into a simplicial complex. Each n-simplicial complex acts like a manager of a business unit: it receives updates from its superiors and transmits information back to its $n + 1$ subsimplicial complexes that are its subordinates. To ensure this simplicial generative AI organization behaves coherently, GAIA builds on the mathematics of the higher-order category theory of simplicial sets and objects. Computations in GAIA, from query answering to foundation model building, are posed in terms of lifting diagrams over simplicial objects. The problem of machine learning in GAIA is modeled as "horn" extensions of simplicial sets: each sub-simplicial complex tries to update its parameters in such a way that a lifting diagram is solved. Traditional approaches used in generative AI using backpropagation can be used to solve "inner" horn extension problems, but addressing "outer horn" extensions requires a more elaborate framework.

At the top level, GAIA uses the simplicial category of ordinal numbers with objects defined as $[n], n \geqslant 0$ and arrows defined as weakly order-preserving mappings $f : [n] \to [m]$, where $f(i) \leqslant f(j), i \leqslant j$. This top-level structure can be viewed as a combinatorial "factory" for constructing,



Graduate Texts in Mathematics

Saunders Mac Lane

Categories for the Working Mathematician

Second Edition

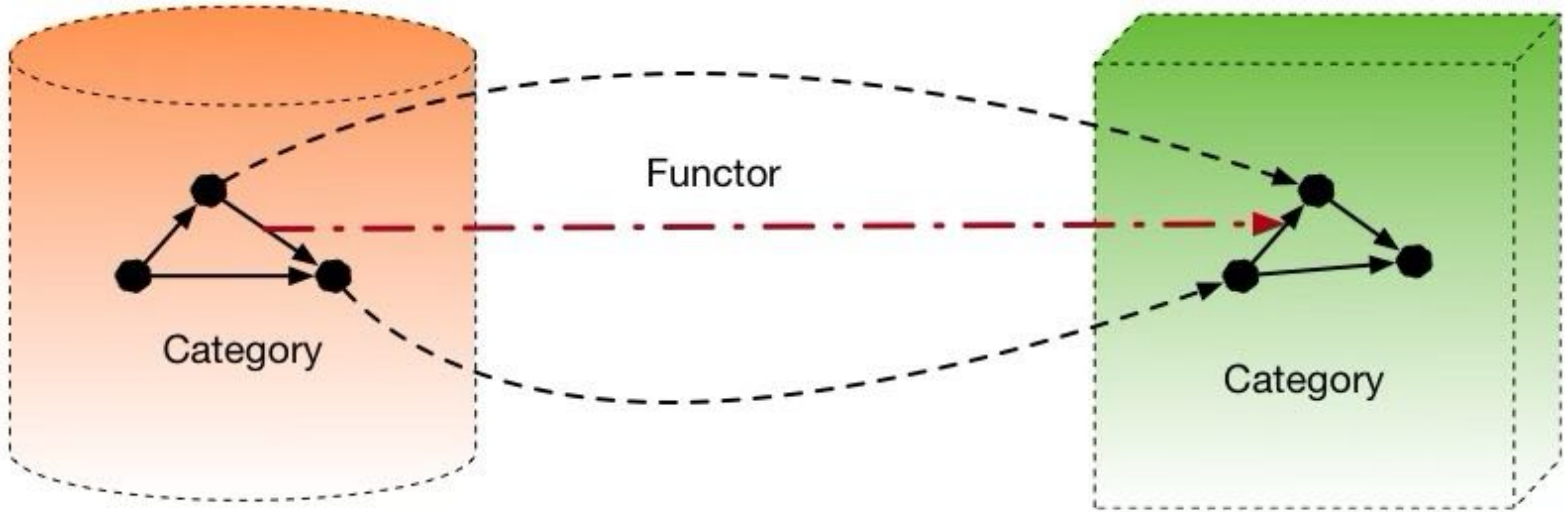
Springer

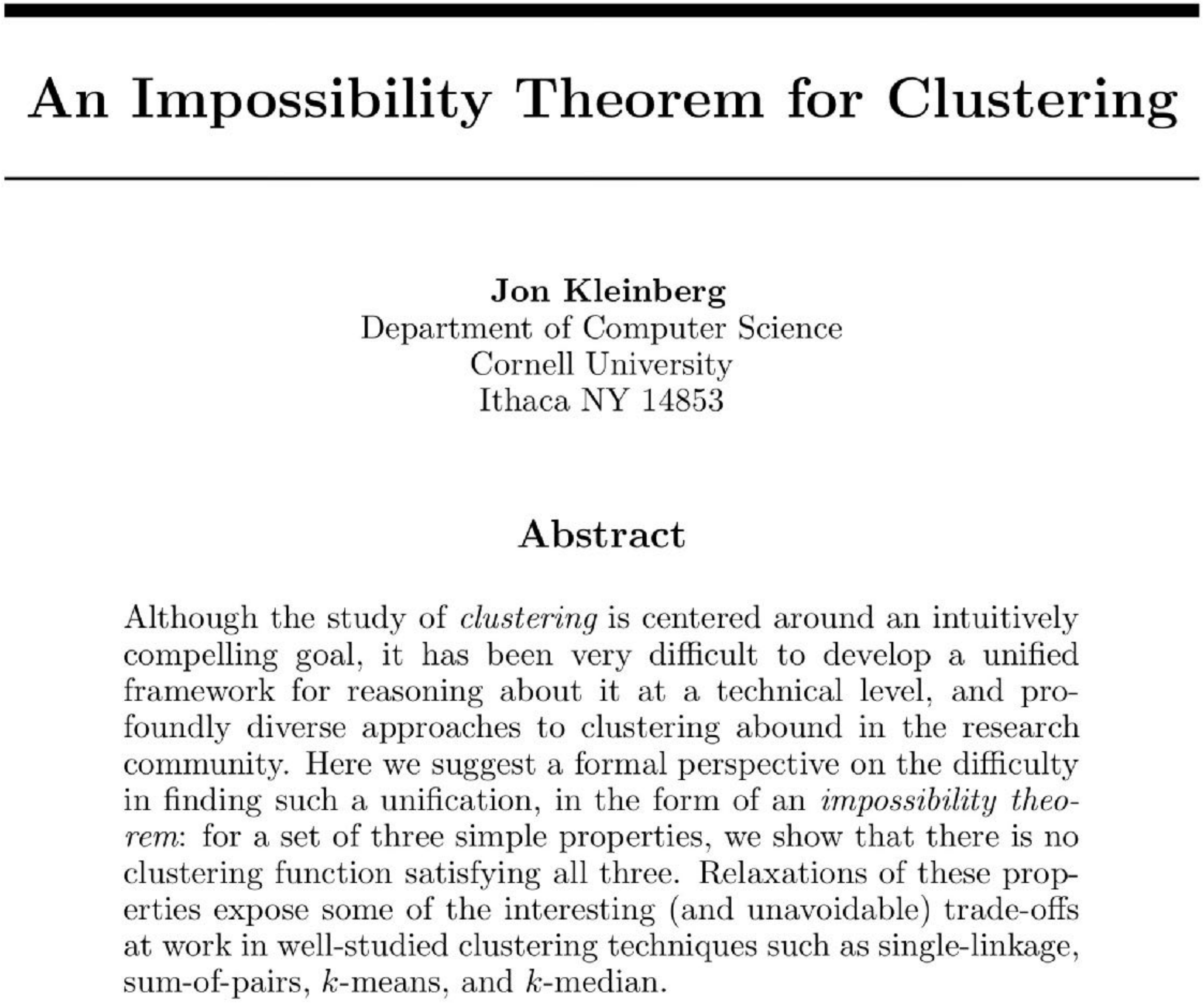
Unified field theory of math!

One formalism that explains it all!

Categories are directed graphs!

Even more basic than set theory


Functor
Category
Category



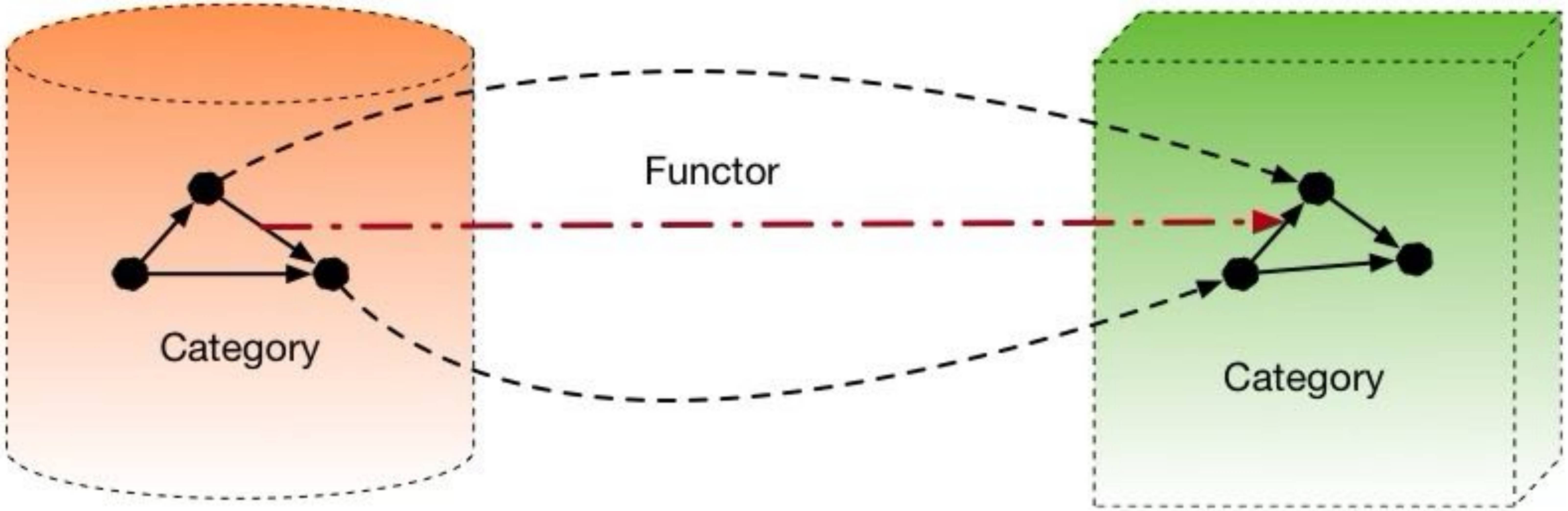
An Impossibility Theorem for Clustering

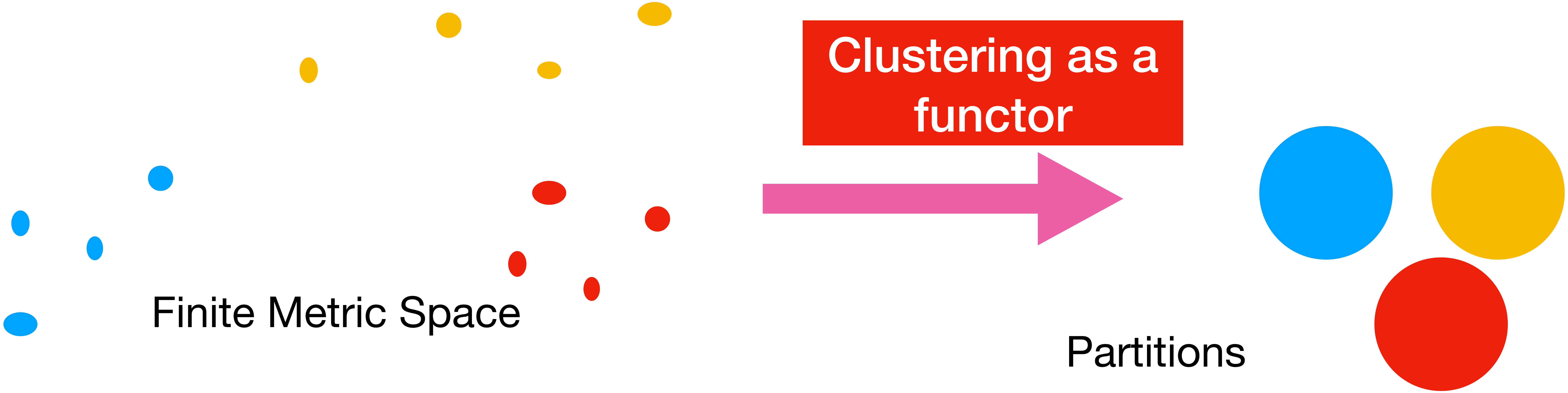
Jon Kleinberg
Department of Computer Science
Cornell University
Ithaca NY 14853

Abstract

Although the study of *clustering* is centered around an intuitively compelling goal, it has been very difficult to develop a unified framework for reasoning about it at a technical level, and profoundly diverse approaches to clustering abound in the research community. Here we suggest a formal perspective on the difficulty in finding such a unification, in the form of an *impossibility theorem*: for a set of three simple properties, we show that there is no clustering function satisfying all three. Relaxations of these properties expose some of the interesting (and unavoidable) trade-offs at work in well-studied clustering techniques such as single-linkage, sum-of-pairs, k-means, and k-median.

- Three properties
  - Scale invariance
  - Monotonicity
  - Surjectivity





Clustering as a functor

Finite Metric Space

Partitions

Journal of Machine Learning Research 11 (2010) 1425-1470 Submitted 4/09; Revised 12/09; Published 4/10

Characterization, Stability and Convergence of Hierarchical Clustering Methods

Gunnar Carlsson GUNNAR@MATH.STANFORD.EDU
Facundo Mémoli* MEMOLI@MATH.STANFORD.EDU
Department of Mathematics
Stanford University
Stanford, CA 94305

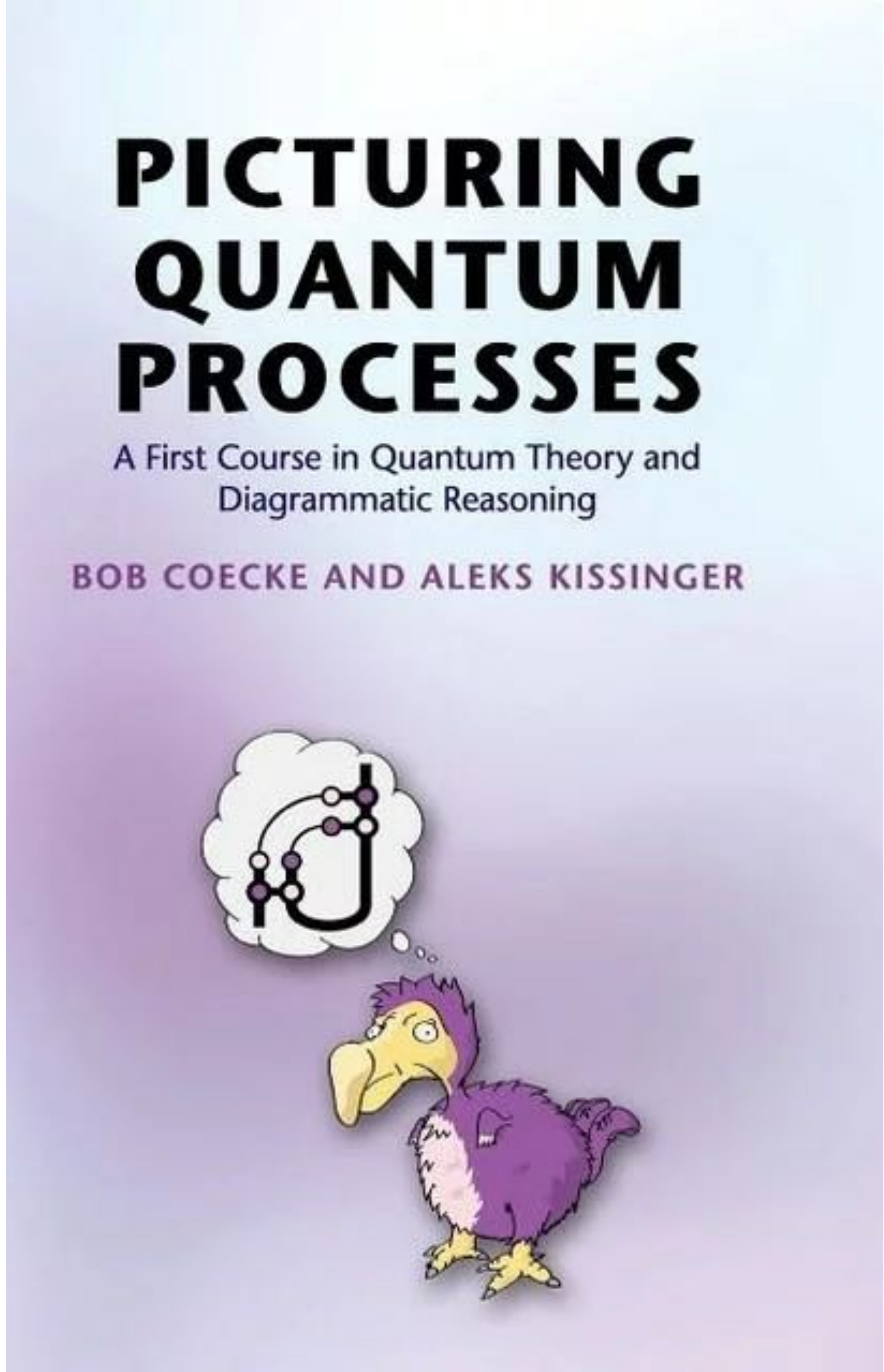
Editor: Ulrike von Luxburg

Abstract

We study hierarchical clustering schemes under an axiomatic view. We show that within this framework, one can prove a theorem analogous to one of Kleinberg (2002), in which one obtains an existence and uniqueness theorem instead of a non-existence result. We explore further properties of this unique scheme: stability and convergence are established. We represent dendrograms as ultrametric spaces and use tools from metric geometry, namely the Gromov-Hausdorff distance, to quantify the degree to which perturbations in the input metric space affect the result of hierarchical methods.

Keywords: clustering, hierarchical clustering, stability of clustering, Gromov-Hausdorff distance

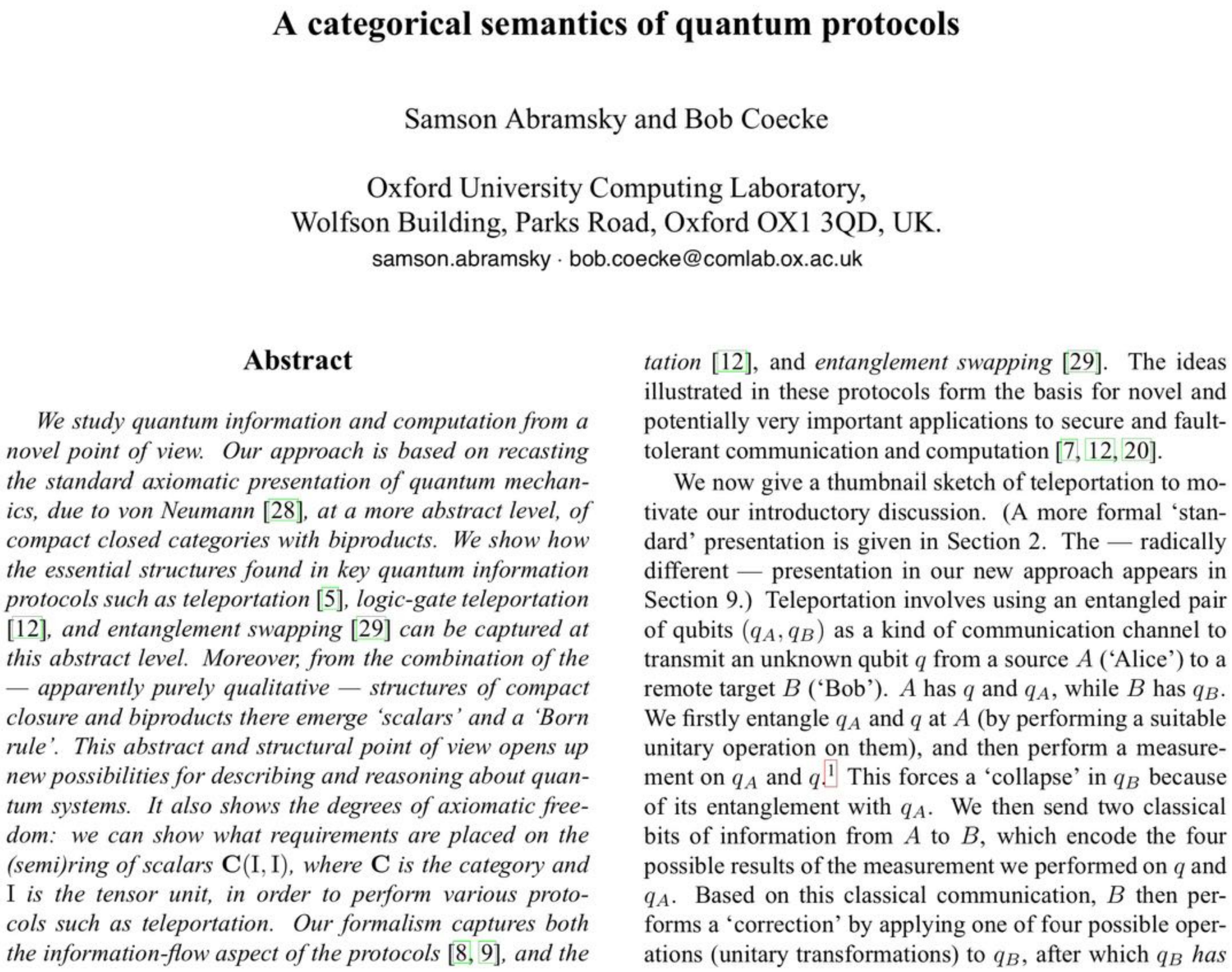
Quantum Computing in Categories



PICTURING QUANTUM PROCESSES

A First Course in Quantum Theory and Diagrammatic Reasoning

BOB COECKE AND ALEKS KISSINGER



A categorical semantics of quantum protocols

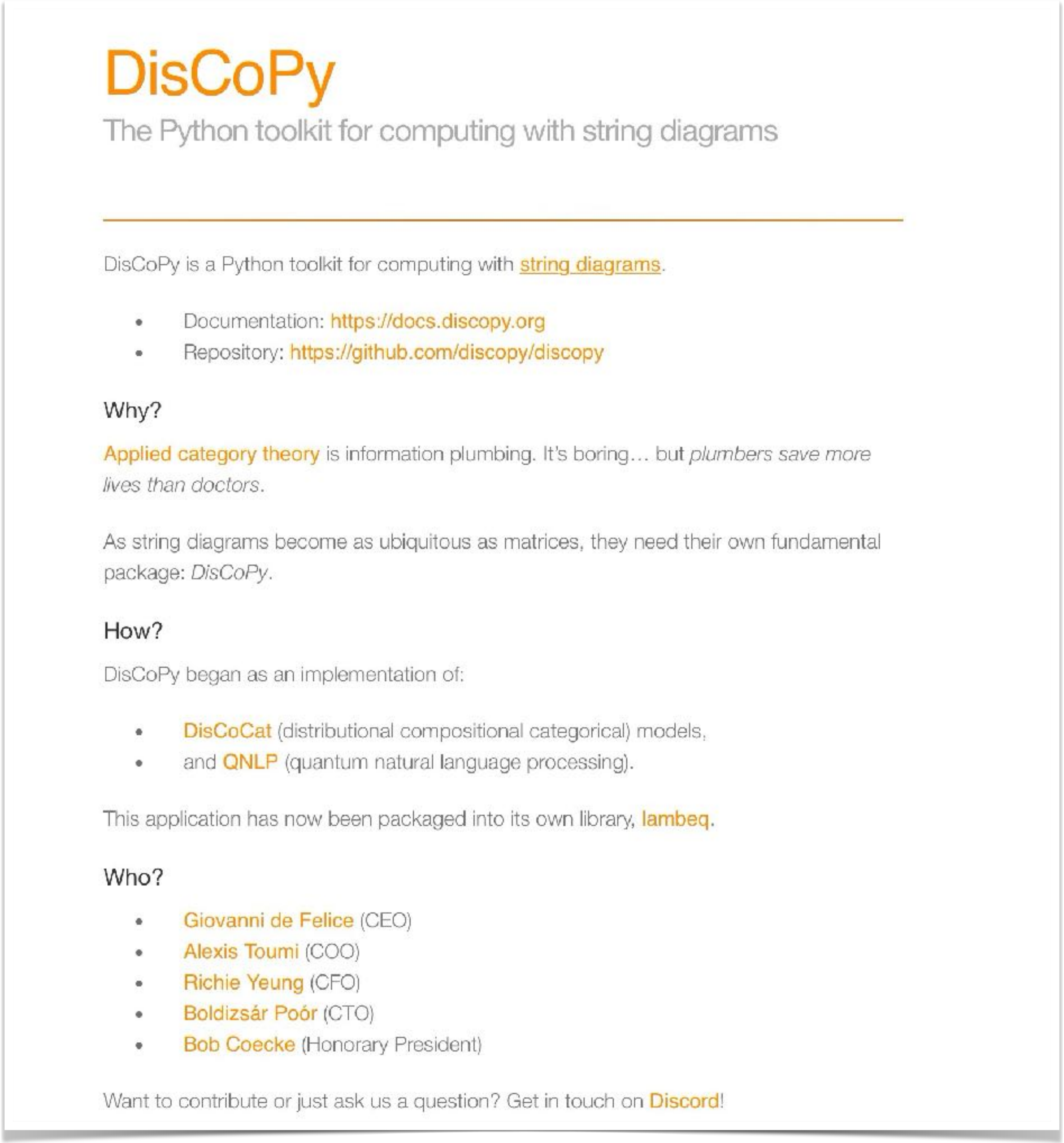
Samson Abramsky and Bob Coecke

Oxford University Computing Laboratory,
Wolfson Building, Parks Road, Oxford OX1 3QD, UK.
samson.abramsky · bob.coecke@comlab.ox.ac.uk

Abstract

We study quantum information and computation from a novel point of view. Our approach is based on recasting the standard axiomatic presentation of quantum mechanics, due to von Neumann [28]*, at a more abstract level, of compact closed categories with biproducts. We show how the essential structures found in key quantum information protocols such as teleportation* [5]*, logic-gate teleportation* [12]*, and entanglement swapping* [29] *can be captured at this abstract level. Moreover, from the combination of the — apparently purely qualitative — structures of compact closure and biproducts there emerge 'scalars' and a 'Born rule'. This abstract and structural point of view opens up new possibilities for describing and reasoning about quantum systems. It also shows the degrees of axiomatic freedom: we can show what requirements are placed on the (semi)ring of scalars* $\mathbf{C}(\mathrm{I},\mathrm{I})$*, where* $\mathbf{C}$ *is the category and* I *is the tensor unit, in order to perform various protocols such as teleportation. Our formalism captures both the information-flow aspect of the protocols* [8, 9]*, and the* *tation* [12]*, and* *entanglement swapping* [29]. The ideas illustrated in these protocols form the basis for novel and potentially very important applications to secure and fault-tolerant communication and computation [7, 12, 20].

We now give a thumbnail sketch of teleportation to motivate our introductory discussion. (A more formal 'standard' presentation is given in Section 2. The — radically different — presentation in our new approach appears in Section 9.) Teleportation involves using an entangled pair of qubits (q_A, q_B) as a kind of communication channel to transmit an unknown qubit q from a source A ('Alice') to a remote target B ('Bob'). A has q and q_A, while B has q_B. We firstly entangle q_A and q at A (by performing a suitable unitary operation on them), and then perform a measurement on q_A and q.[1] This forces a 'collapse' in q_B because of its entanglement with q_A. We then send two classical bits of information from A to B, which encode the four possible results of the measurement we performed on q and q_A. Based on this classical communication, B then performs a 'correction' by applying one of four possible operations (unitary transformations) to q_B, after which q_B has



DisCoPy

The Python toolkit for computing with string diagrams

DisCoPy is a Python toolkit for computing with string diagrams.

- Documentation: https://docs.discopy.org
- Repository: https://github.com/discopy/discopy

Why?

Applied category theory is information plumbing. It's boring… but *plumbers save more lives than doctors*.

As string diagrams become as ubiquitous as matrices, they need their own fundamental package: *DisCoPy*.

How?

DisCoPy began as an implementation of:

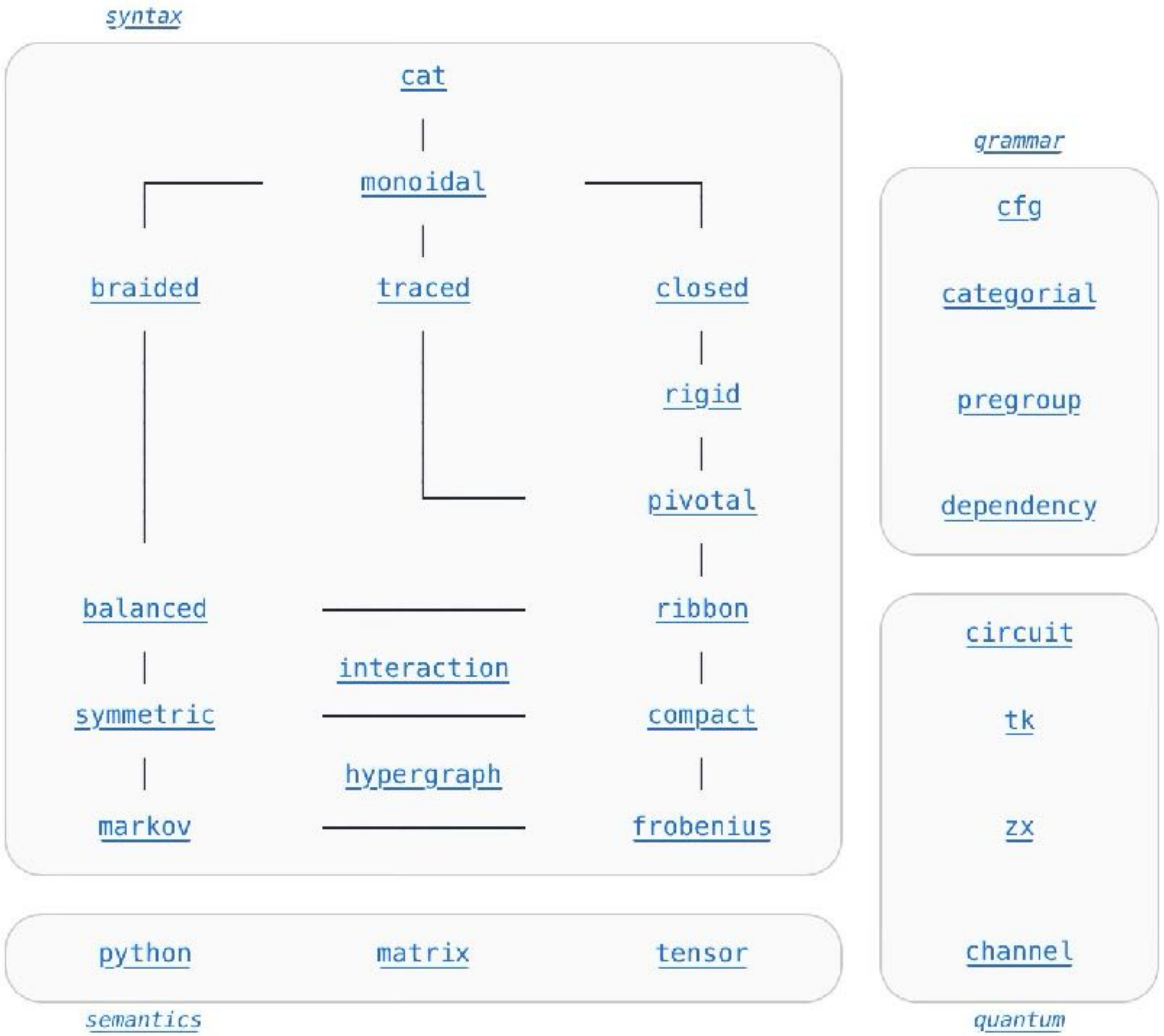
- DisCoCat (distributional compositional categorical) models,
- and QNLP (quantum natural language processing).

This application has now been packaged into its own library, lambeq.

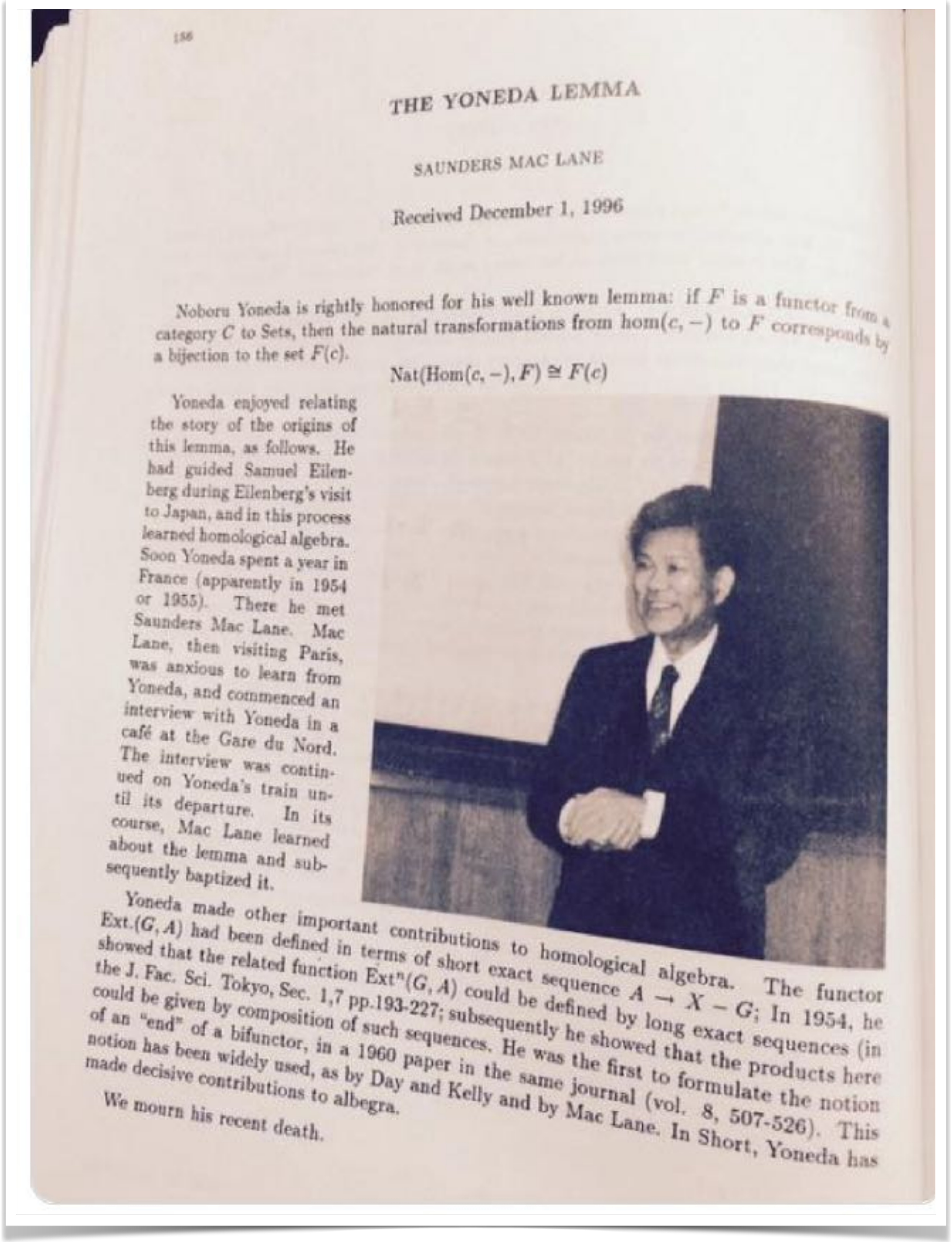
Who?

- Giovanni de Felice (CEO)
- Alexis Toumi (COO)
- Richie Yeung (CFO)
- Boldizsár Poór (CTO)
- Bob Coecke (Honorary President)

Want to contribute or just ask us a question? Get in touch on Discord!



Le Lemme du Gare du Nord



THE YONEDA LEMMA

SAUNDERS MAC LANE

Received December 1, 1996

Noboru Yoneda is rightly honored for his well known lemma: if F is a functor from a category C to Sets, then the natural transformations from $\hom(c, -)$ to F corresponds by a bijection to the set $F(c)$.

$$\mathrm{Nat}(\mathrm{Hom}(c, -), F) \cong F(c)$$

Yoneda enjoyed relating the story of the origins of this lemma, as follows. He had guided Samuel Eilenberg during Eilenberg's visit to Japan, and in this process learned homological algebra. Soon Yoneda spent a year in France (apparently in 1954 or 1955). There he met Saunders Mac Lane. Mac Lane, then visiting Paris, was anxious to learn from Yoneda, and commenced an interview with Yoneda in a café at the Gare du Nord. The interview was continued on Yoneda's train until its departure. In its course, Mac Lane learned about the lemma and subsequently baptized it.

Yoneda made other important contributions to homological algebra. The functor $\mathrm{Ext}.(G, A)$ had been defined in terms of short exact sequence $A \to X - G$; In 1954, he showed that the related function $\mathrm{Ext}^n(G, A)$ could be defined by long exact sequences (in the J. Fac. Sci. Tokyo, Sec. 1,7 pp.193-227; subsequently he showed that the products here could be given by composition of such sequences. He was the first to formulate the notion of an "end" of a bifunctor, in a 1960 paper in the same journal (vol. 8, 507-526). This notion has been widely used, as by Day and Kelly and by Mac Lane. In Short, Yoneda has made decisive contributions to albegra.

We mourn his recent death.

$$\mathrm{Hom}(C(-, x), F) \simeq Fx$$

The Yoneda Lemma came to "life" in 1954

Coincidentally, Turing died in 1954

Natural Transformations

$$\begin{tikzcd}
C \arrow[r, bend left, "F"] \arrow[r, bend right, "G"] & D
\end{tikzcd}$$

$$\begin{tikzcd}
Fc \arrow[r, "Ff"] \arrow[d, "\mu_c"'] & Fc' \arrow[d, "\mu_{c'}"] \\
Gc \arrow[r, "Gf"] & Gc'
\end{tikzcd}$$

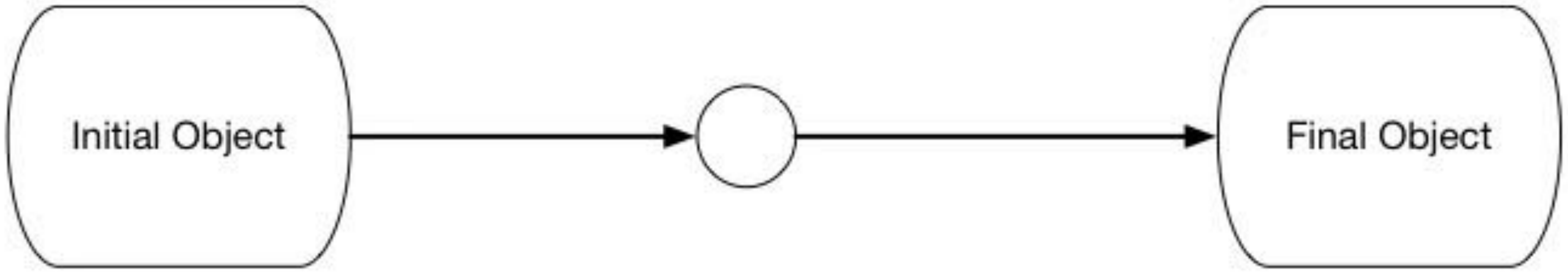
Natural Transformations for Deep Learning

Category of Parameters

$$\begin{tikzcd}
C \arrow[r, bend left, "\text{Backpropagation}"] \arrow[r, bend right, "\text{Stochastic approximation}"'] & D
\end{tikzcd}$$

Category of Learners

$$\begin{tikzcd}
Fc \arrow[r, "Ff"] \arrow[d, "\mu_c"'] & Fc' \arrow[d, "\mu_{c'}"] \\
Gc \arrow[r, "Gf"] & Gc'
\end{tikzcd}$$



A covariant functor F is representable iff its category of elements $\int F \simeq \int C(c,-) \simeq c/C$ has an initial object

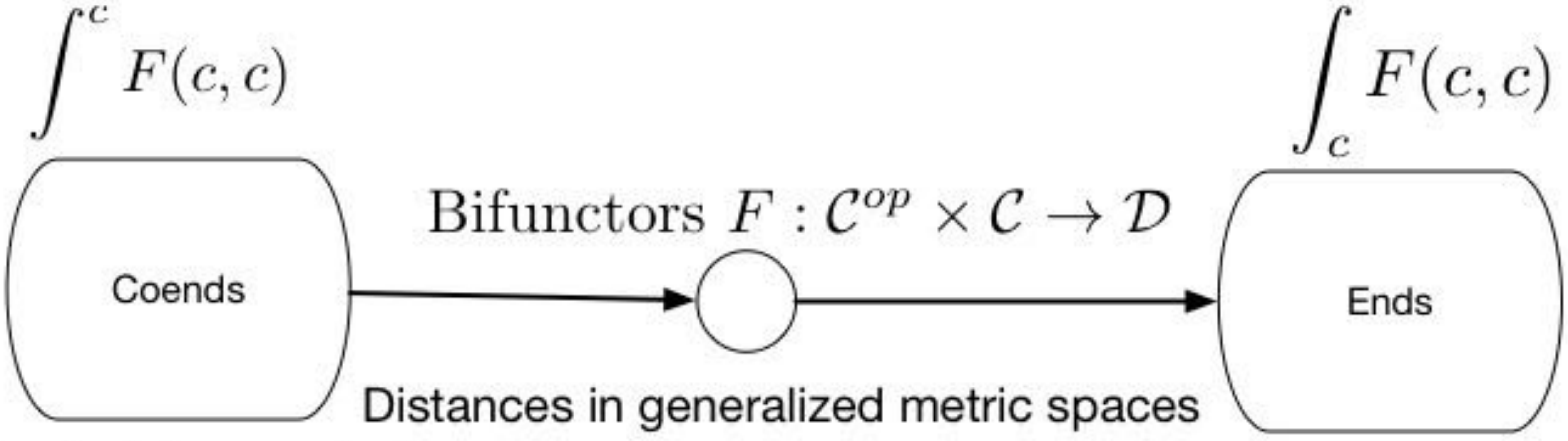
A contravariant functor F is representable iff its category of elements $\int F \simeq \int C(-,c)$ has a final object

Category of Elements: For any set-valued functor $F : C \rightarrow \mathbf{Set}$

Objects: (c, x), where c is a category object, and x is an element of Fc

Arrows: (c, x) —> (c’, y) where f: c -> c’ is a morphism in C so that F(f)(x) = y (covariant)

Arrows: (c, x) —> (c’, y) and f: c-> c’ is a morphism in C so that F(f)(y) = x (contravariant)



$\int^{c} F(c,c)$

Coends

Bifunctors $F : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{D}$

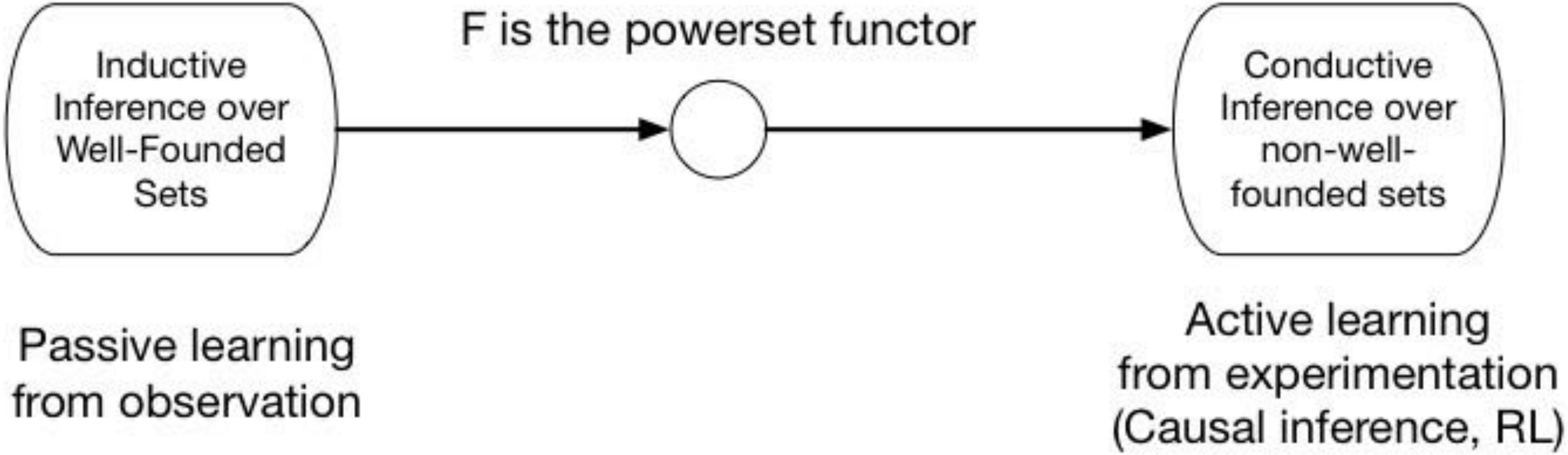
Distances in generalized metric spaces

$\int_{c} F(c,c)$

Ends

Topological data analysis
Manifold learning

Probabilistic Generative Models



F is the powerset functor

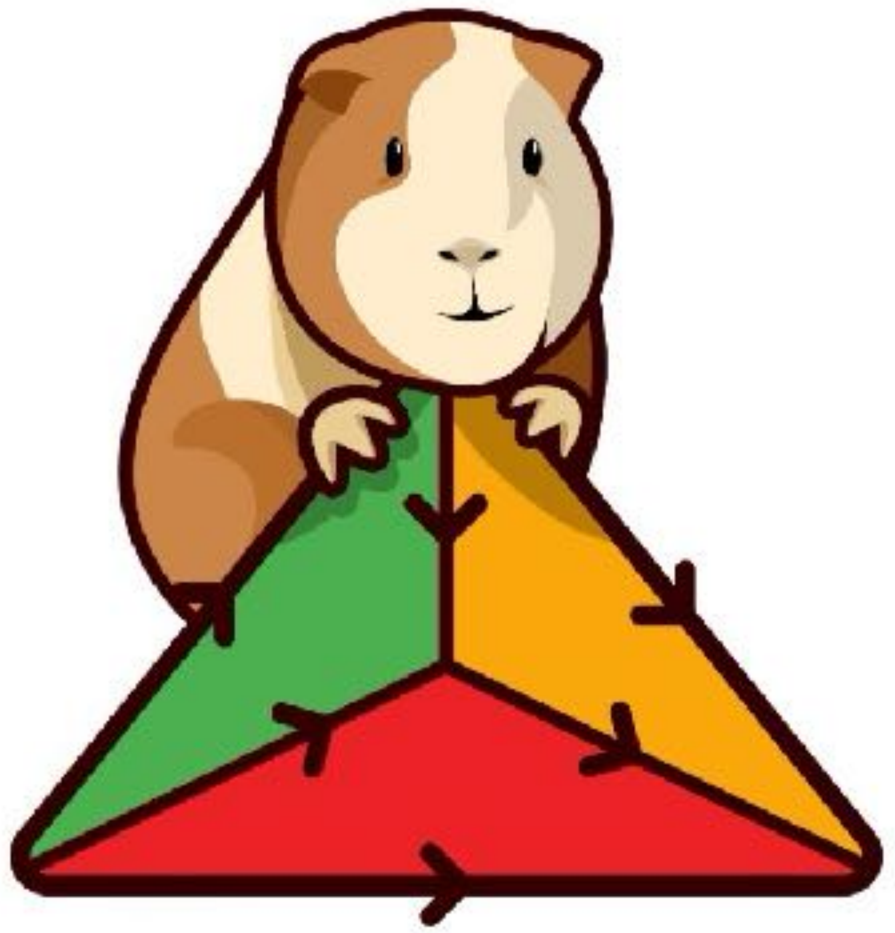
Inductive Inference over Well-Founded Sets

Conductive Inference over non-well-founded sets

Passive learning
from observation

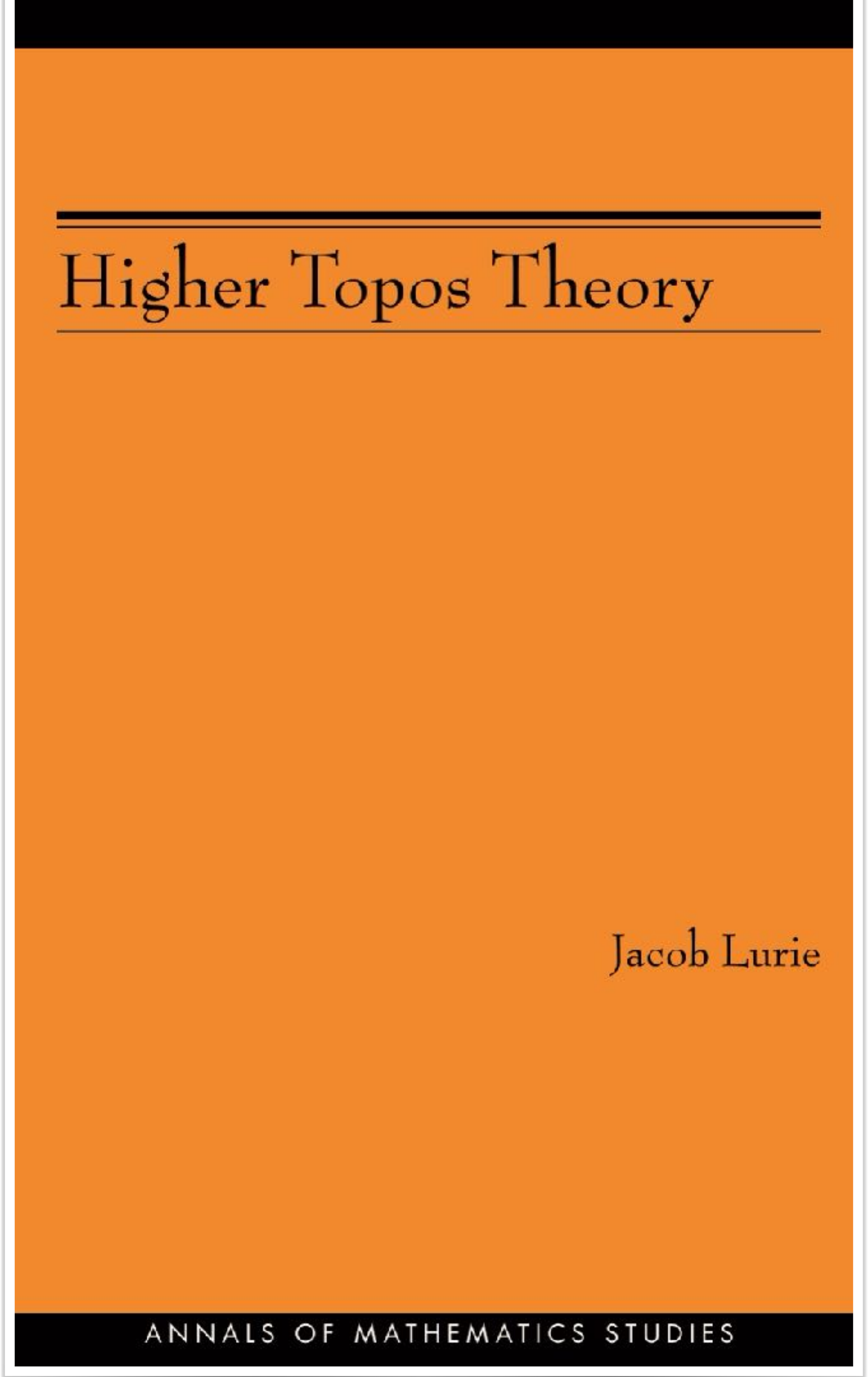
Active learning
from experimentation
(Causal inference, RL)

GAIA: A Higher-Order Categorical Framework for Generative AI



Kerodon

an online resource for homotopy-coherent mathematics



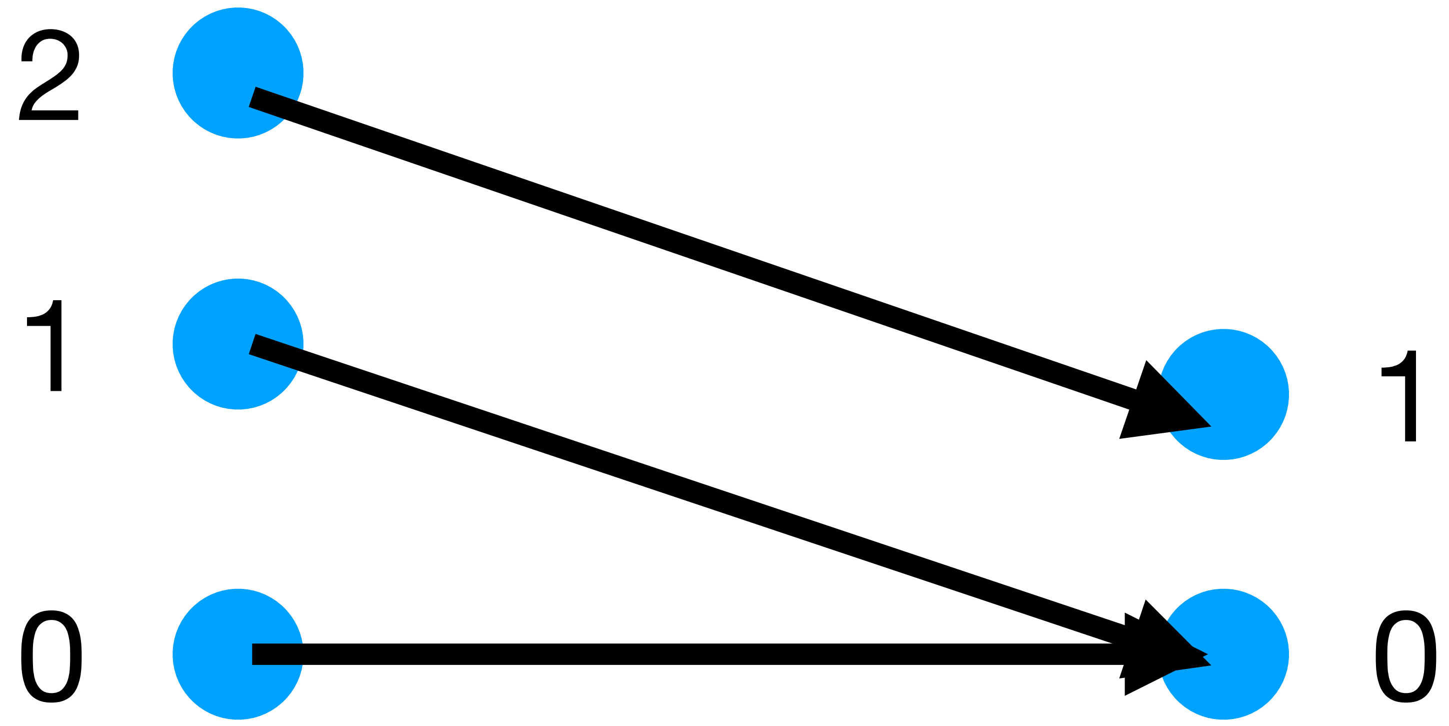
Higher Topos Theory

Jacob Lurie

ANNALS OF MATHEMATICS STUDIES

Simplicial Category Δ

- **Objects**: ordinal numbers
  - $[n] = \{0,1,\ldots,n-1\}$
- **Arrows**:
  - $f : [m] \to [n]$
  - If $i \leq j$, then $f(i) \leq f(j)$
  - All morphisms can be built out of primitive injections/surjections
    - $\delta_i : [n] \to [n+1]$: injection skipping i
    - $\sigma_i : [n] \to [n-1]$, surjection repeating i



Simplicial Sets: Contravariant Functors

$$\begin{tikzcd}
0 \arrow[r] & 1 \arrow[r, shift left] \arrow[r, shift right] & 2 \arrow[r, shift left=2] \arrow[r] \arrow[r, shift right=2] & 3
\end{tikzcd}$$

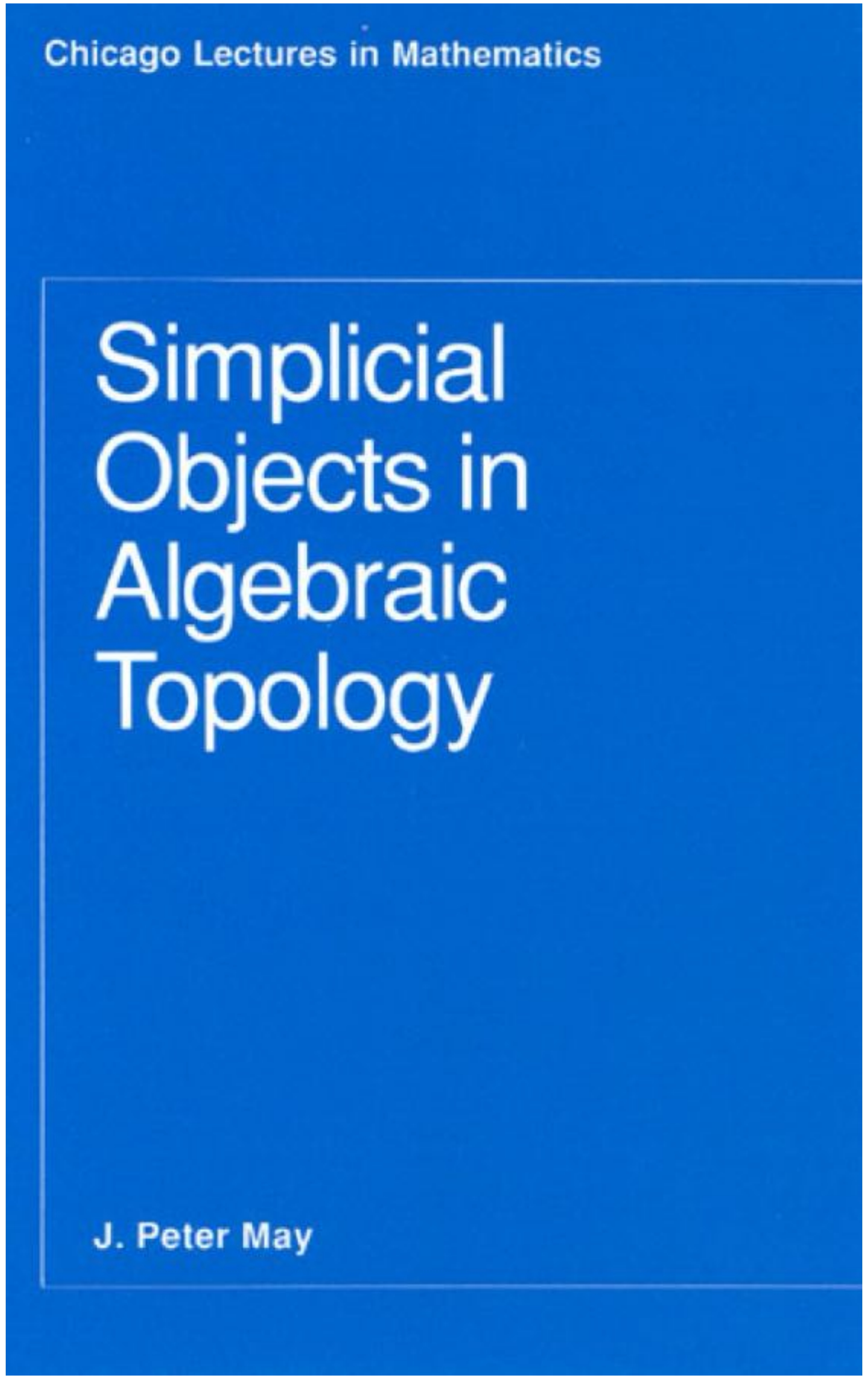
$\delta_i^n : [n] \to [n+1]$

$$\begin{tikzcd}
0 & 1 \arrow[l] & 2 \arrow[l] & 3 \arrow[l, shift left] \arrow[l, shift right]
\end{tikzcd}$$

$\sigma_i^n : [n+1] \to [n]$

$$X_n : [n] \to X : \Delta^{op} \to X$$

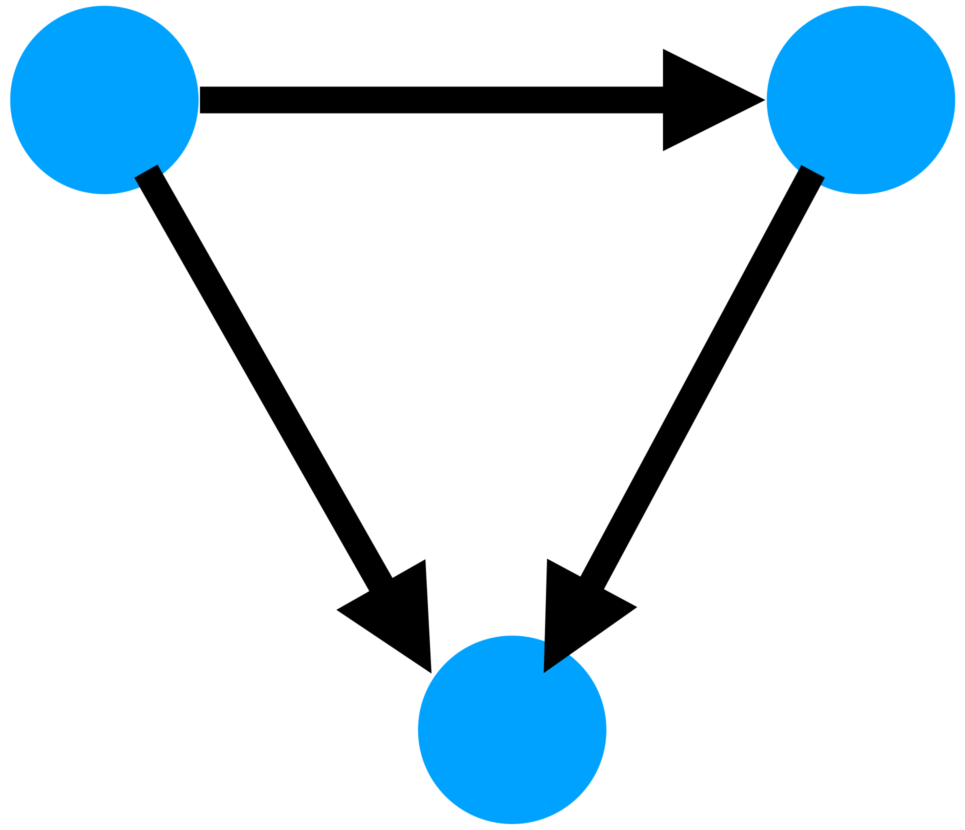
Simplicial Objects: One stop ML shopping center

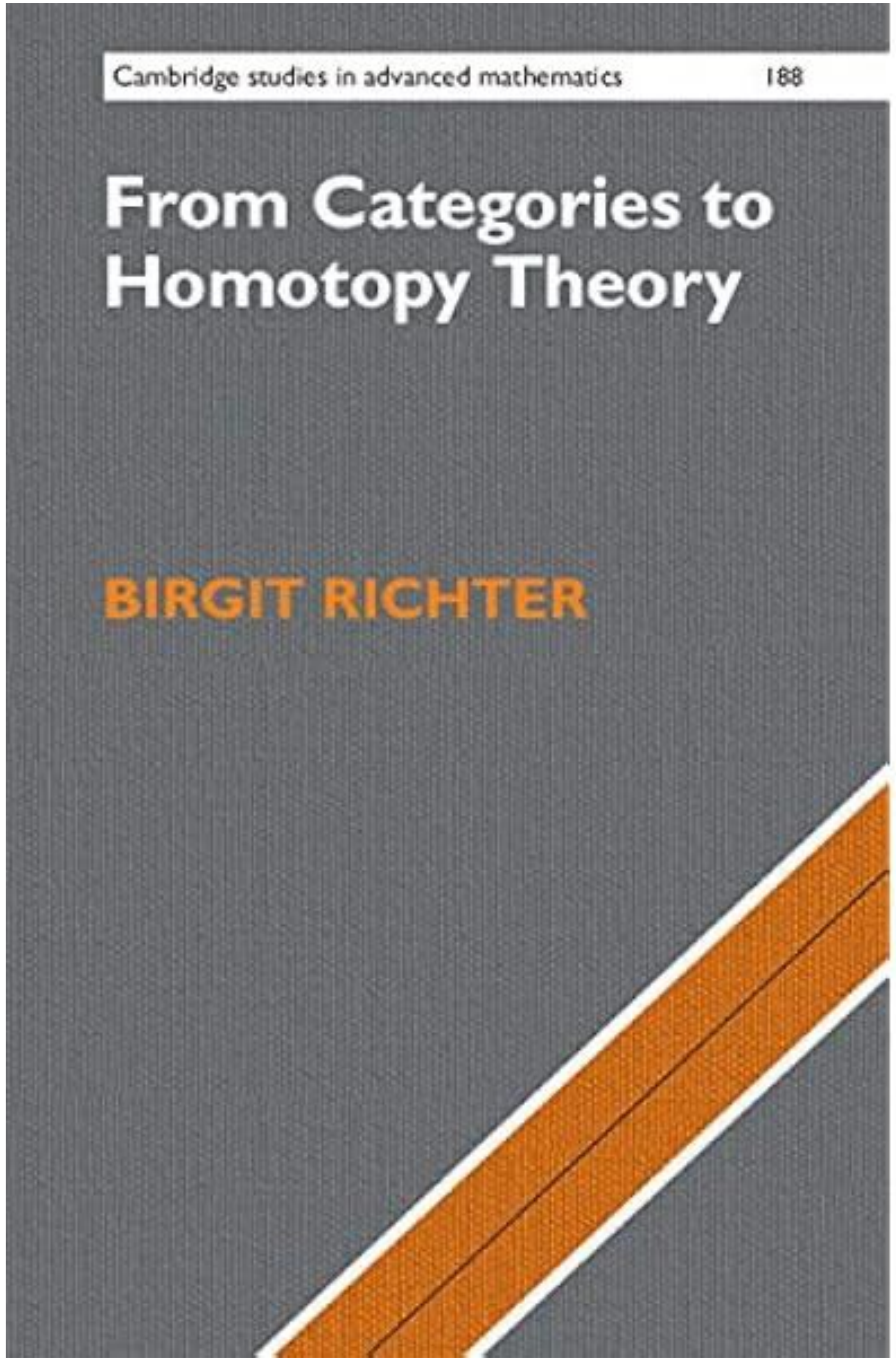


Chicago Lectures in Mathematics

Simplicial Objects in Algebraic Topology

J. Peter May

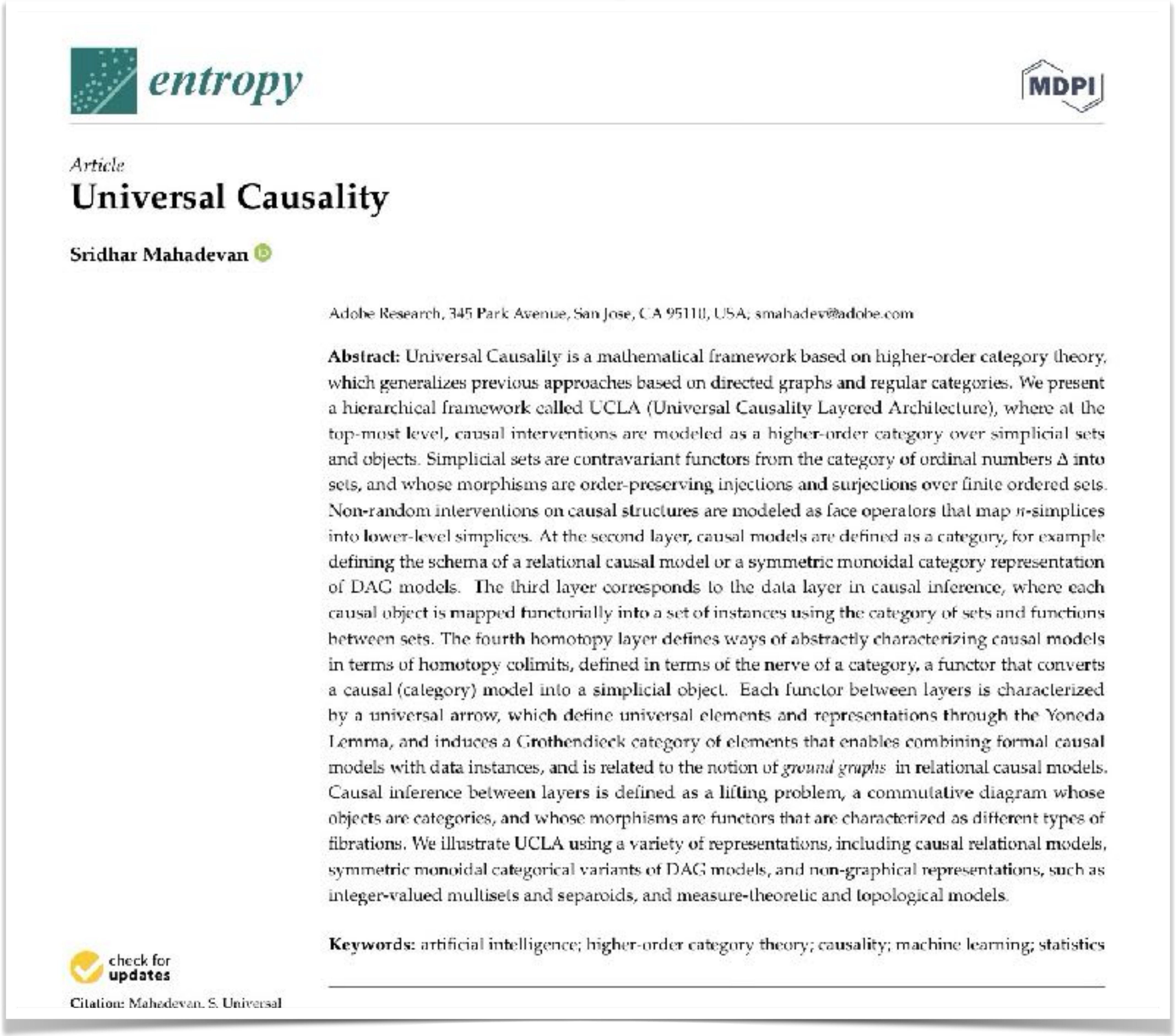




Cambridge studies in advanced mathematics 188

From Categories to Homotopy Theory

BIRGIT RICHTER



entropy

MDPI

Article

Universal Causality

Sridhar Mahadevan

Adobe Research, 345 Park Avenue, San Jose, CA 95110, USA; smahadev@adobe.com

Abstract: Universal Causality is a mathematical framework based on higher-order category theory, which generalizes previous approaches based on directed graphs and regular categories. We present a hierarchical framework called UCLA (Universal Causality Layered Architecture), where at the top-most level, causal interventions are modeled as a higher-order category over simplicial sets and objects. Simplicial sets are contravariant functors from the category of ordinal numbers Δ into sets, and whose morphisms are order-preserving injections and surjections over finite ordered sets. Non-random interventions on causal structures are modeled as face operators that map n-simplices into lower-level simplices. At the second layer, causal models are defined as a category, for example defining the schema of a relational causal model or a symmetric monoidal category representation of DAG models. The third layer corresponds to the data layer in causal inference, where each causal object is mapped functorially into a set of instances using the category of sets and functions between sets. The fourth homotopy layer defines ways of abstractly characterizing causal models in terms of homotopy colimits, defined in terms of the nerve of a category, a functor that converts a causal (category) model into a simplicial object. Each functor between layers is characterized by a universal arrow, which define universal elements and representations through the Yoneda Lemma, and induces a Grothendieck category of elements that enables combining formal causal models with data instances, and is related to the notion of *ground graphs* in relational causal models. Causal inference between layers is defined as a lifting problem, a commutative diagram whose objects are categories, and whose morphisms are functors that are characterized as different types of fibrations. We illustrate UCLA using a variety of representations, including causal relational models, symmetric monoidal categorical variants of DAG models, and non-graphical representations, such as integer-valued multisets and separoids, and measure-theoretic and topological models.

Keywords: artificial intelligence; higher-order category theory; causality; machine learning; statistics

check for updates

Citation: Mahadevan, S. Universal

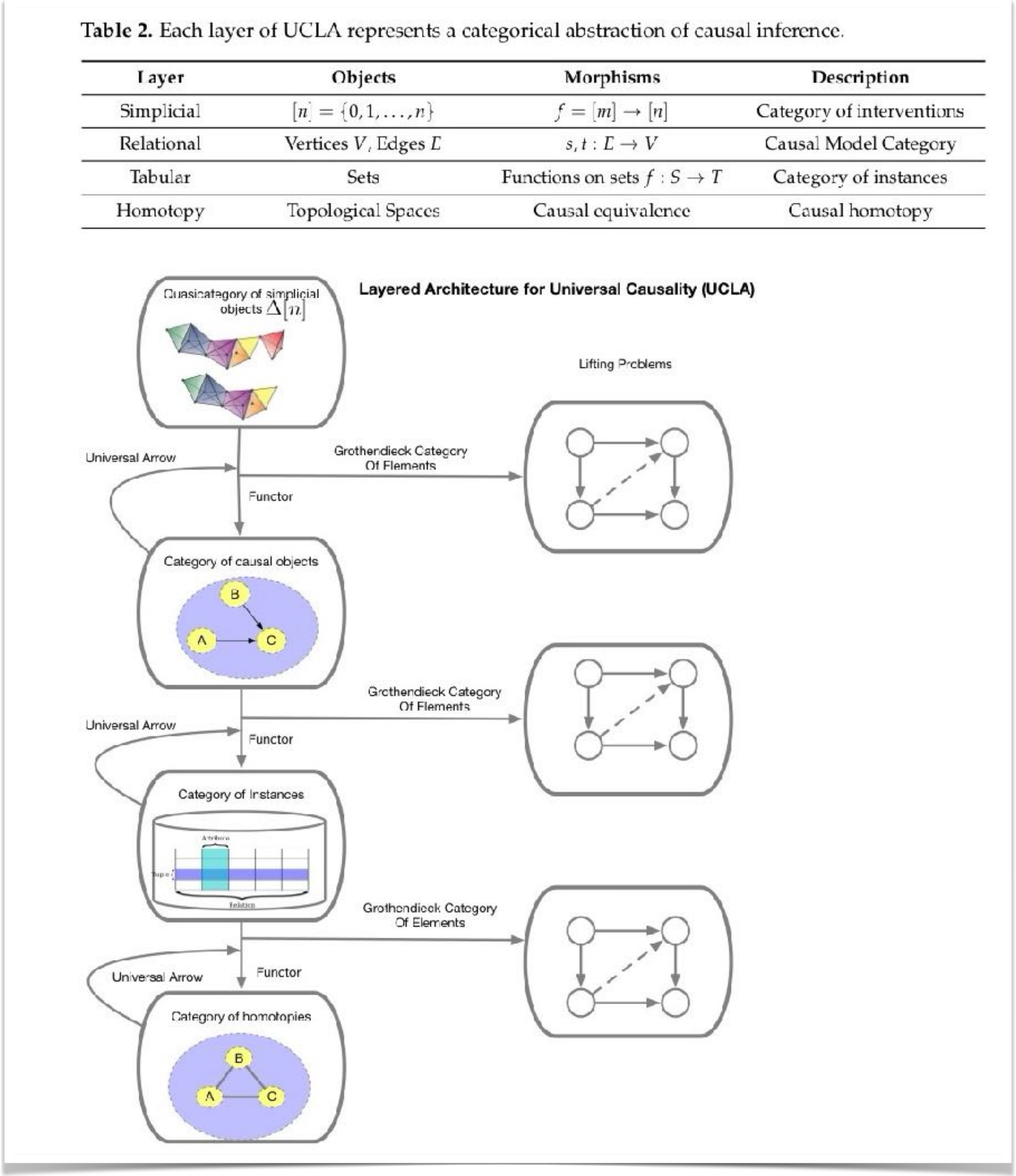
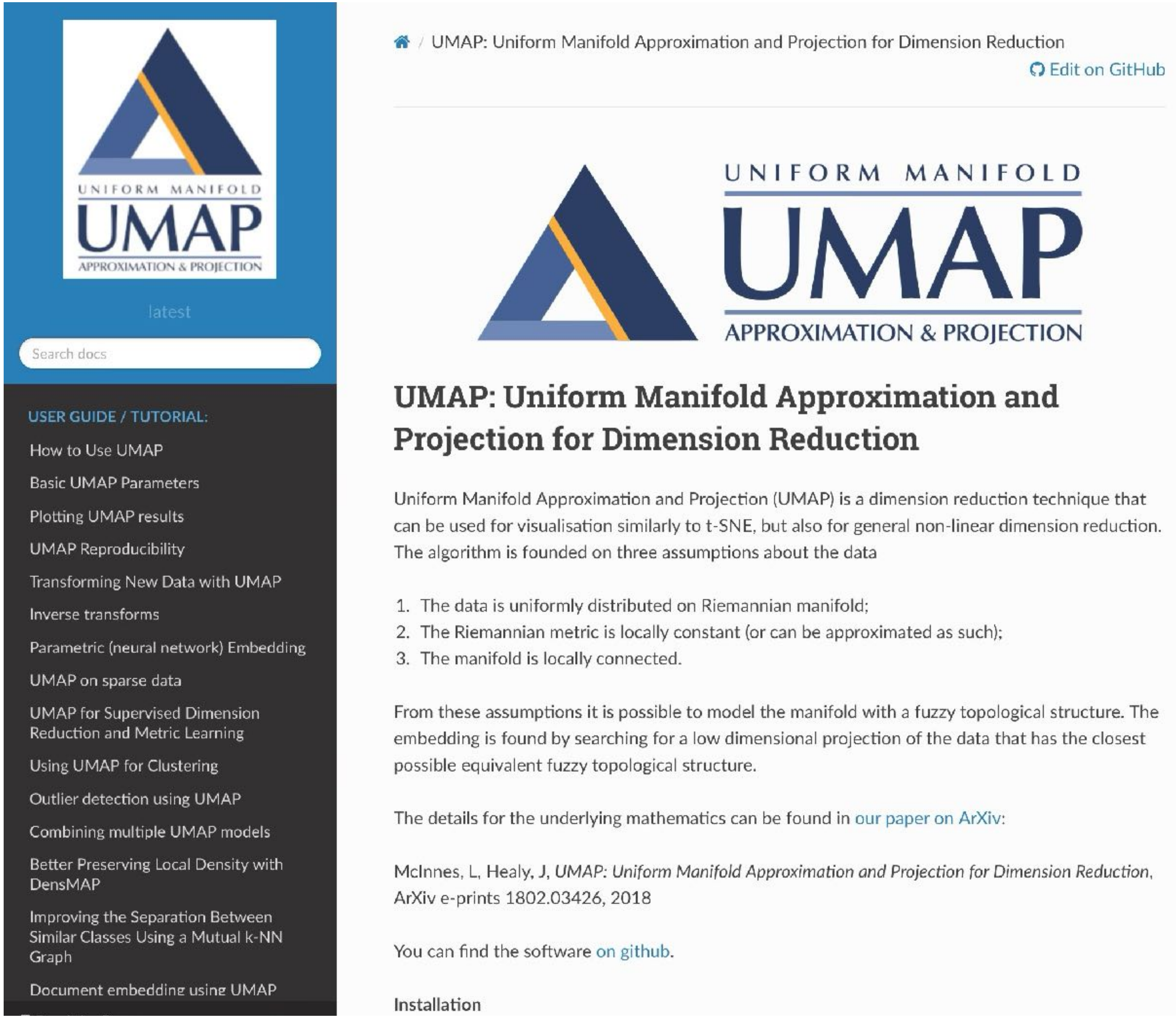


Table 2. Each layer of UCLA represents a categorical abstraction of causal inference.

Layer	Objects	Morphisms	Description
Simplicial	$[n] = \{0, 1, \ldots, n\}$	$f = [m] \to [n]$	Category of interventions
Relational	Vertices V, Edges E	$s, t : E \to V$	Causal Model Category
Tabular	Sets	Functions on sets $f : S \to T$	Category of instances
Homotopy	Topological Spaces	Causal equivalence	Causal homotopy

Layered Architecture for Universal Causality (UCLA)

Quasicategory of simplicial objects $\Delta[n]$
Lifting Problems
Universal Arrow
Grothendieck Category Of Elements
Functor
Category of causal objects
B
A
C
Universal Arrow
Grothendieck Category Of Elements
Functor
Category of Instances
Attribute
Tuple
Relation
Grothendieck Category Of Elements
Universal Arrow
Functor
Category of homotopies
B
A
C


UNIFORM MANIFOLD
UMAP
APPROXIMATION & PROJECTION
latest
Search docs
USER GUIDE / TUTORIAL:
How to Use UMAP
Basic UMAP Parameters
Plotting UMAP results
UMAP Reproducibility
Transforming New Data with UMAP
Inverse transforms
Parametric (neural network) Embedding
UMAP on sparse data
UMAP for Supervised Dimension Reduction and Metric Learning
Using UMAP for Clustering
Outlier detection using UMAP
Combining multiple UMAP models
Better Preserving Local Density with DensMAP
Improving the Separation Between Similar Classes Using a Mutual k-NN Graph
Document embedding using UMAP
/ UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction
Edit on GitHub
UNIFORM MANIFOLD
UMAP
APPROXIMATION & PROJECTION

UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction

Uniform Manifold Approximation and Projection (UMAP) is a dimension reduction technique that can be used for visualisation similarly to t-SNE, but also for general non-linear dimension reduction. The algorithm is founded on three assumptions about the data

1. The data is uniformly distributed on Riemannian manifold;
2. The Riemannian metric is locally constant (or can be approximated as such);
3. The manifold is locally connected.

From these assumptions it is possible to model the manifold with a fuzzy topological structure. The embedding is found by searching for a low dimensional projection of the data that has the closest possible equivalent fuzzy topological structure.

The details for the underlying mathematics can be found in our paper on ArXiv:

McInnes, L, Healy, J, *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*, ArXiv e-prints 1802.03426, 2018

You can find the software on github.

Installation

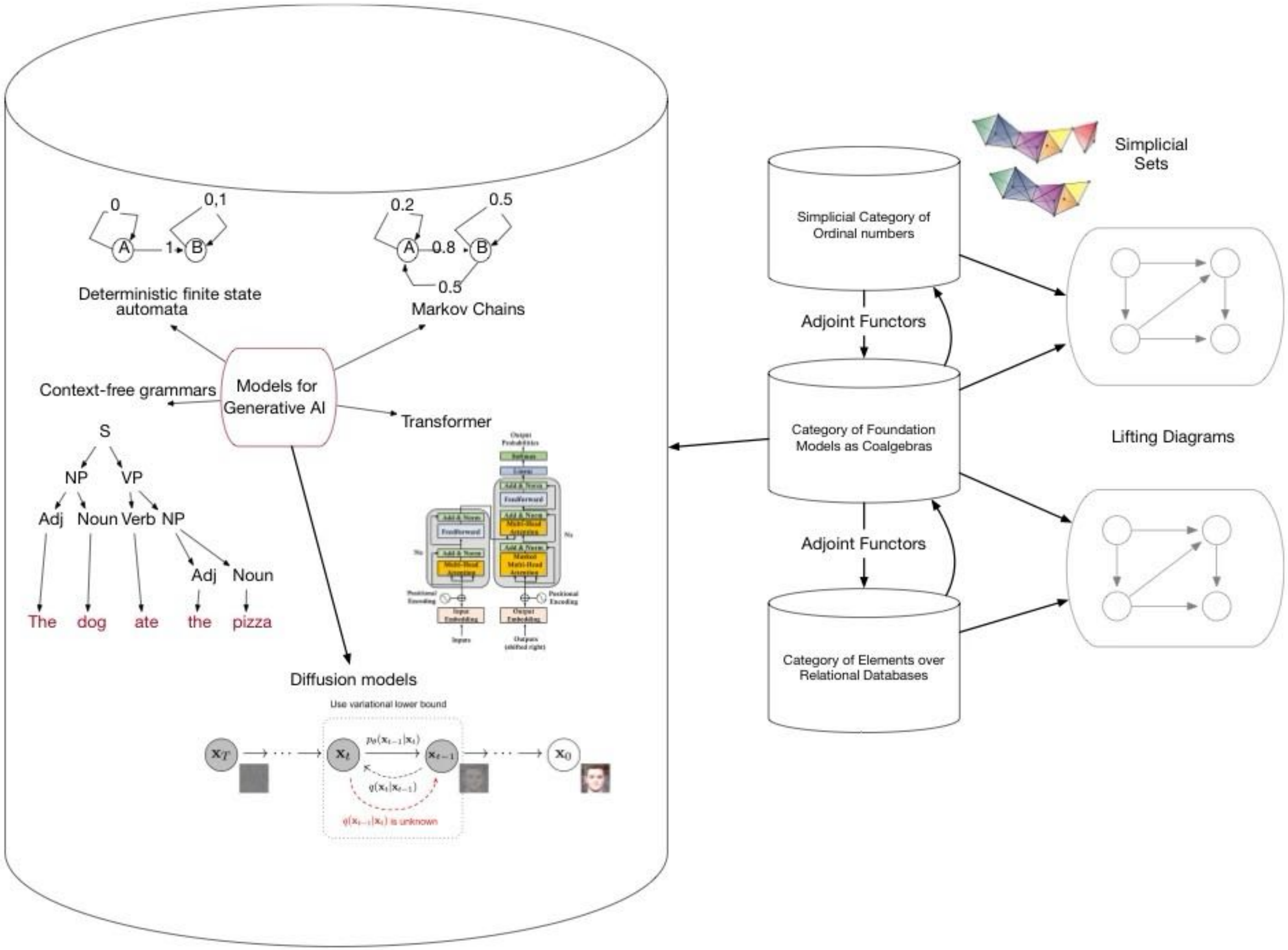
Best data visualization method in ML today

Scalable to millions of data points

Used widely in biology

Based on higher-order category theory of simplicial sets & objects

GAIA: Categorical Foundations of Generative AI



Models for Generative AI

Deterministic finite state automata

Markov Chains

Context-free grammars

Transformer

Diffusion models

Use variational lower bound

Simplicial Sets

Simplicial Category of Ordinal numbers

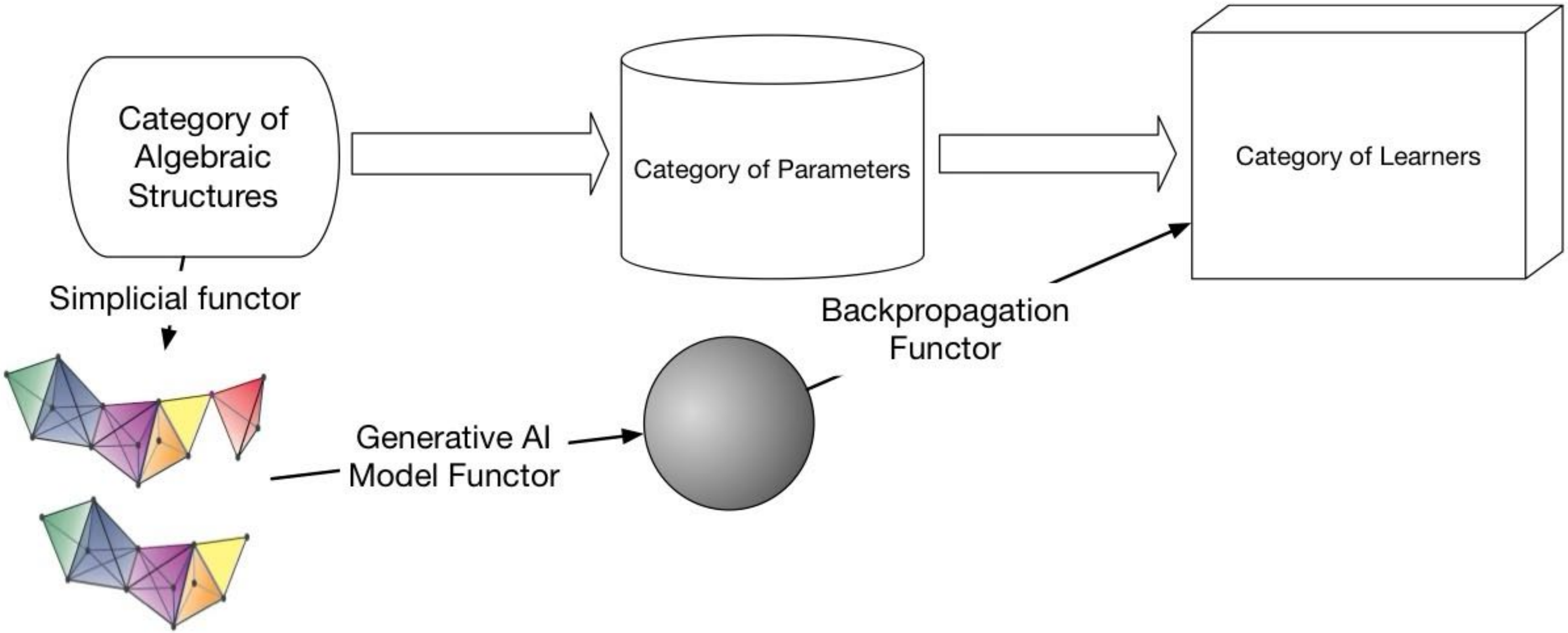
Adjoint Functors

Category of Foundation Models as Coalgebras

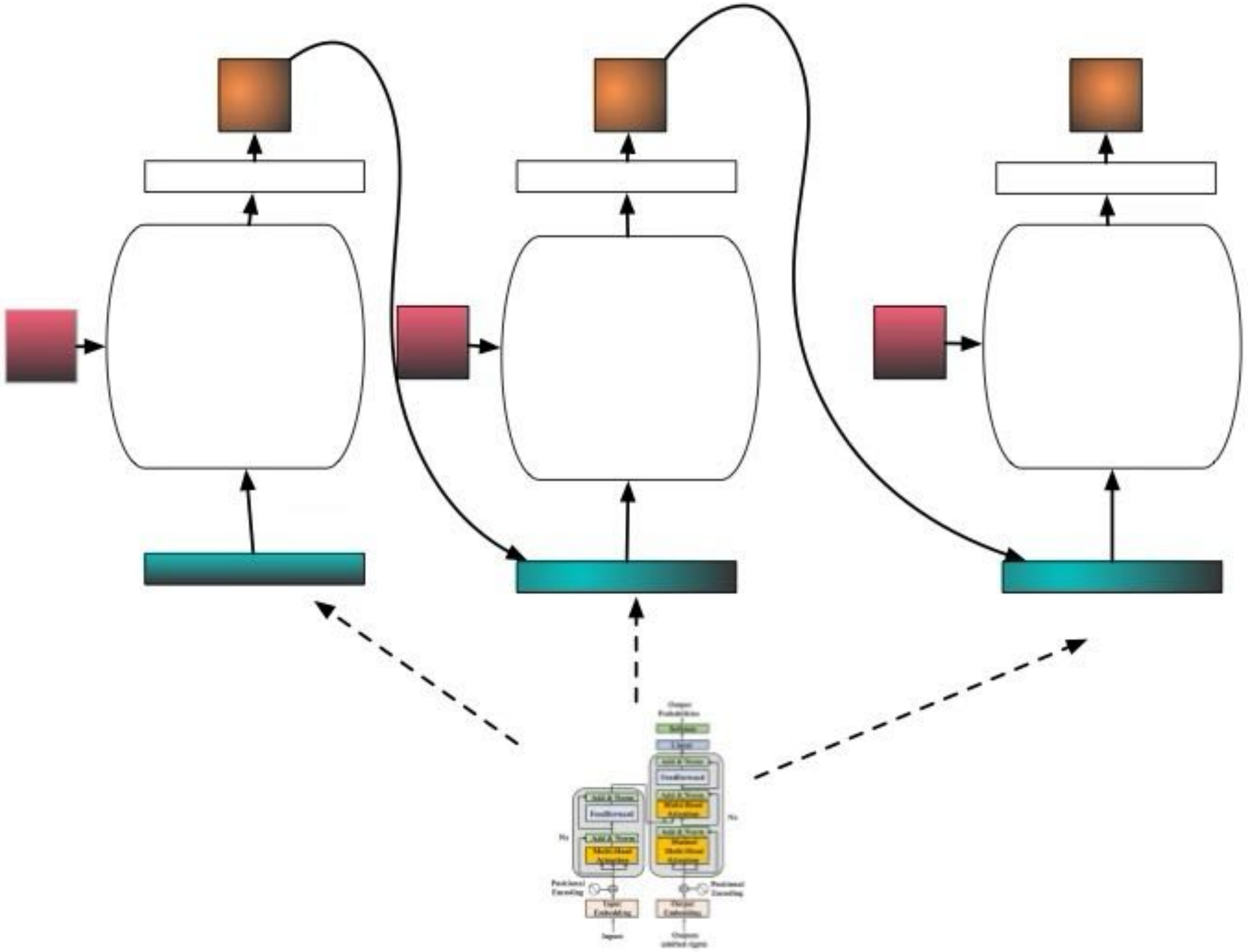
Lifting Diagrams

Adjoint Functors

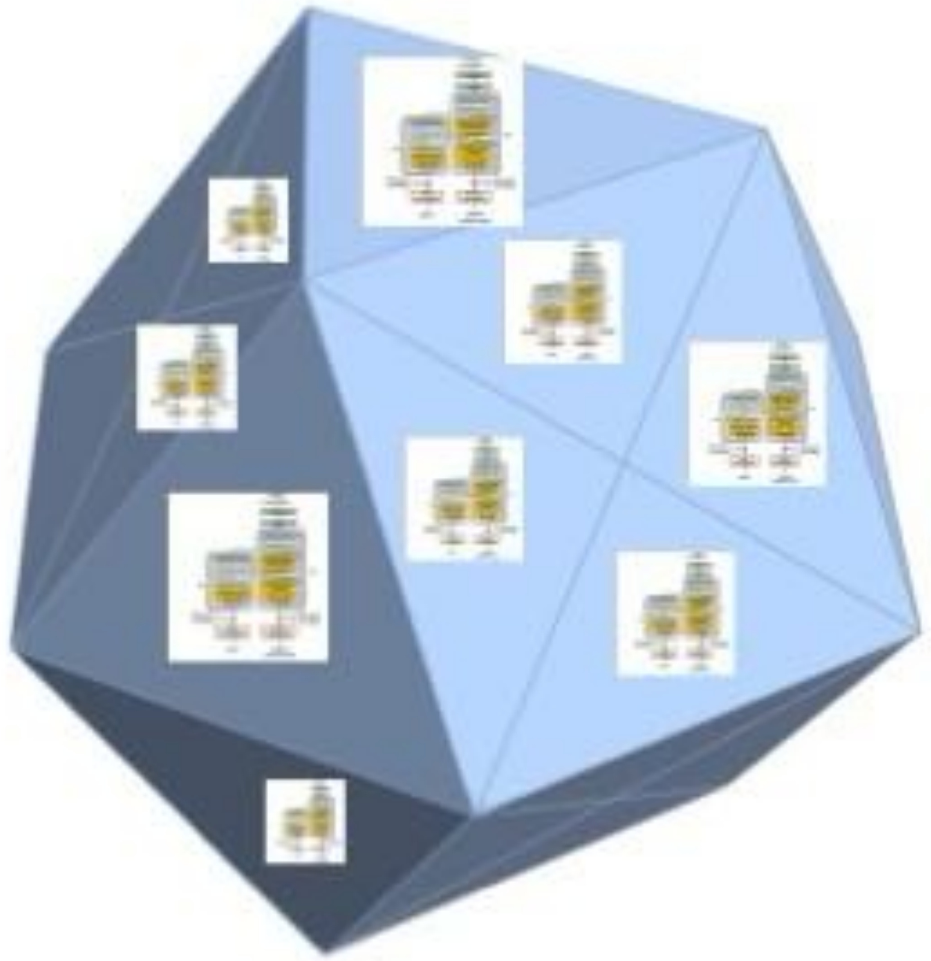
Category of Elements over Relational Databases

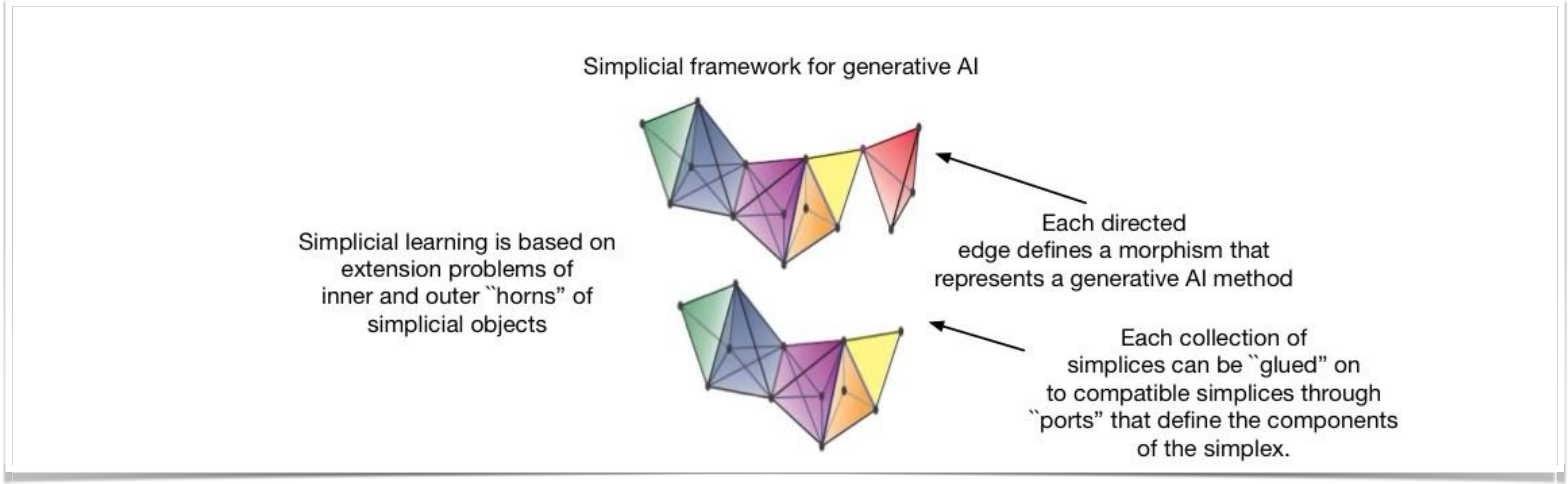

Category of Algebraic Structures
Category of Parameters
Category of Learners
Simplicial functor
Backpropagation Functor
Generative AI Model Functor

Sequential Generative AI Model



Simplicial Complex of Gen AI Models in GAIA



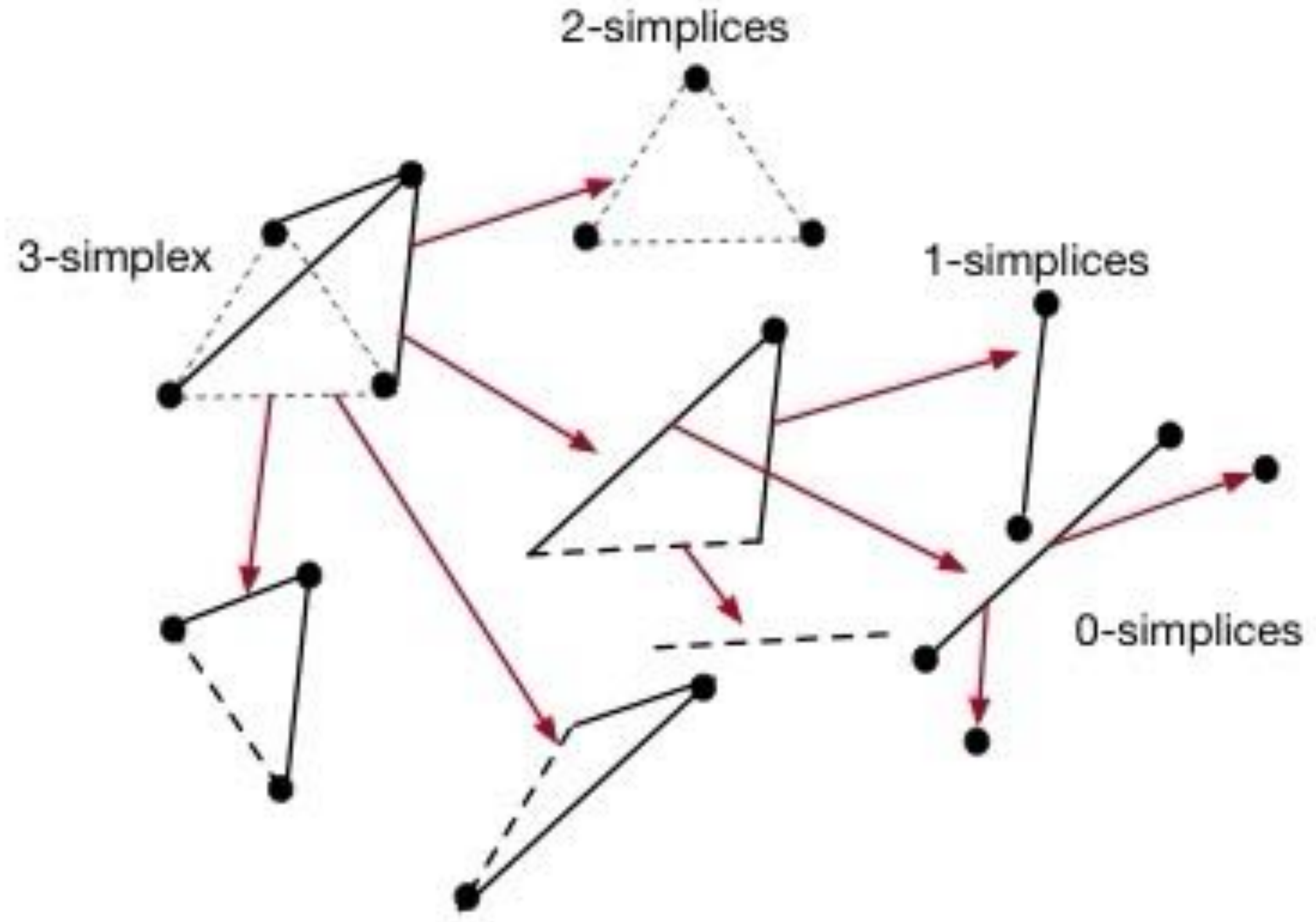


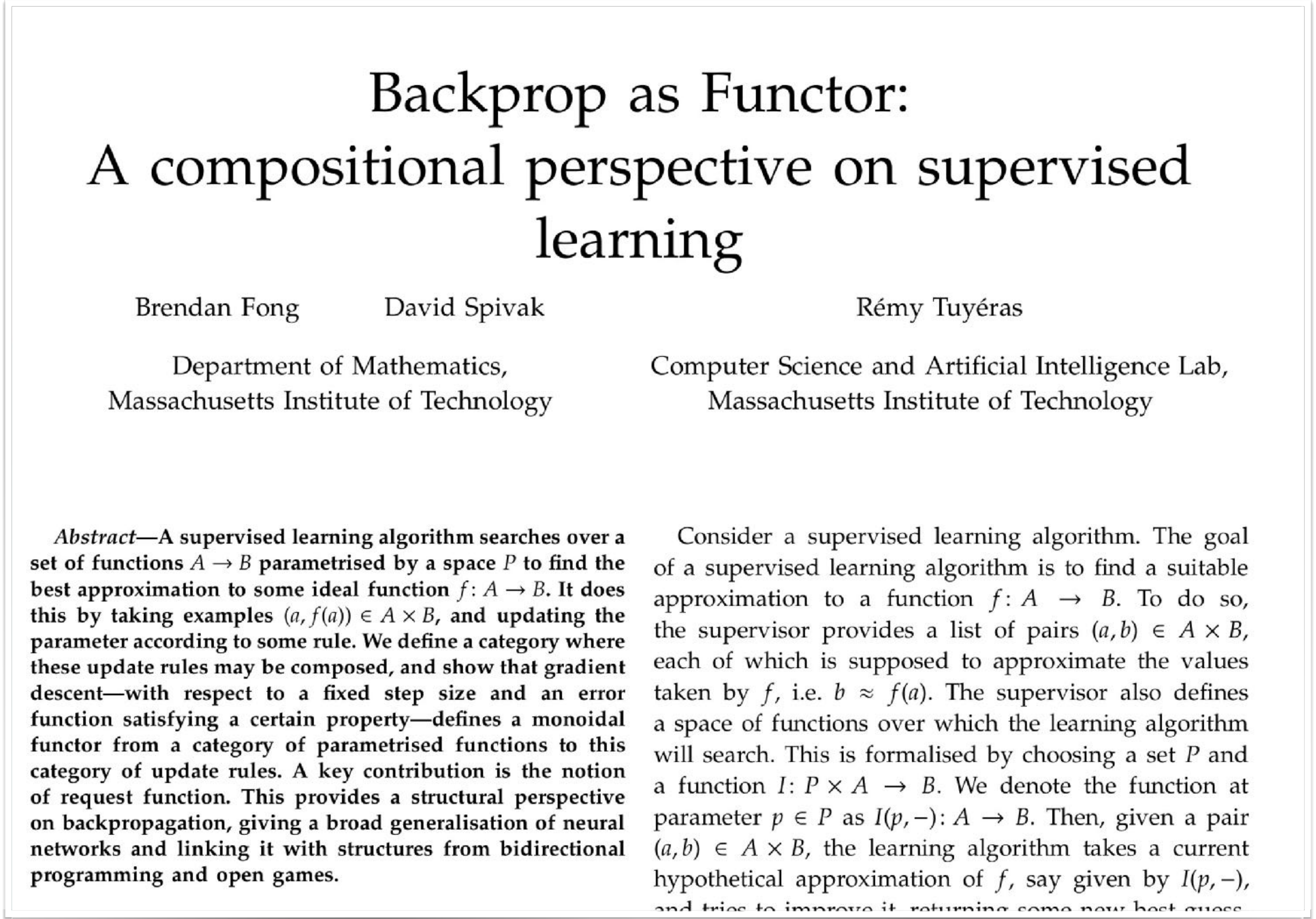
Simplicial framework for generative AI

Simplicial learning is based on extension problems of inner and outer ``horns" of simplicial objects

Each directed edge defines a morphism that represents a generative AI method

Each collection of simplices can be ``glued" on to compatible simplices through ``ports" that define the components of the simplex.





Backprop as Functor: A compositional perspective on supervised learning

Brendan Fong David Spivak Rémy Tuyéras

Department of Mathematics,
Massachusetts Institute of Technology

Computer Science and Artificial Intelligence Lab,
Massachusetts Institute of Technology

***Abstract*—A supervised learning algorithm searches over a set of functions $A \to B$ parametrised by a space P to find the best approximation to some ideal function $f\colon A \to B$. It does this by taking examples $(a, f(a)) \in A \times B$, and updating the parameter according to some rule. We define a category where these update rules may be composed, and show that gradient descent—with respect to a fixed step size and an error function satisfying a certain property—defines a monoidal functor from a category of parametrised functions to this category of update rules. A key contribution is the notion of request function. This provides a structural perspective on backpropagation, giving a broad generalisation of neural networks and linking it with structures from bidirectional programming and open games.**

Consider a supervised learning algorithm. The goal of a supervised learning algorithm is to find a suitable approximation to a function $f\colon A \to B$. To do so, the supervisor provides a list of pairs $(a, b) \in A \times B$, each of which is supposed to approximate the values taken by f, i.e. $b \approx f(a)$. The supervisor also defines a space of functions over which the learning algorithm will search. This is formalised by choosing a set P and a function $I\colon P \times A \to B$. We denote the function at parameter $p \in P$ as $I(p, -)\colon A \to B$. Then, given a pair $(a, b) \in A \times B$, the learning algorithm takes a current hypothetical approximation of f, say given by $I(p, -)$, and tries to improve it, returning some new best guess

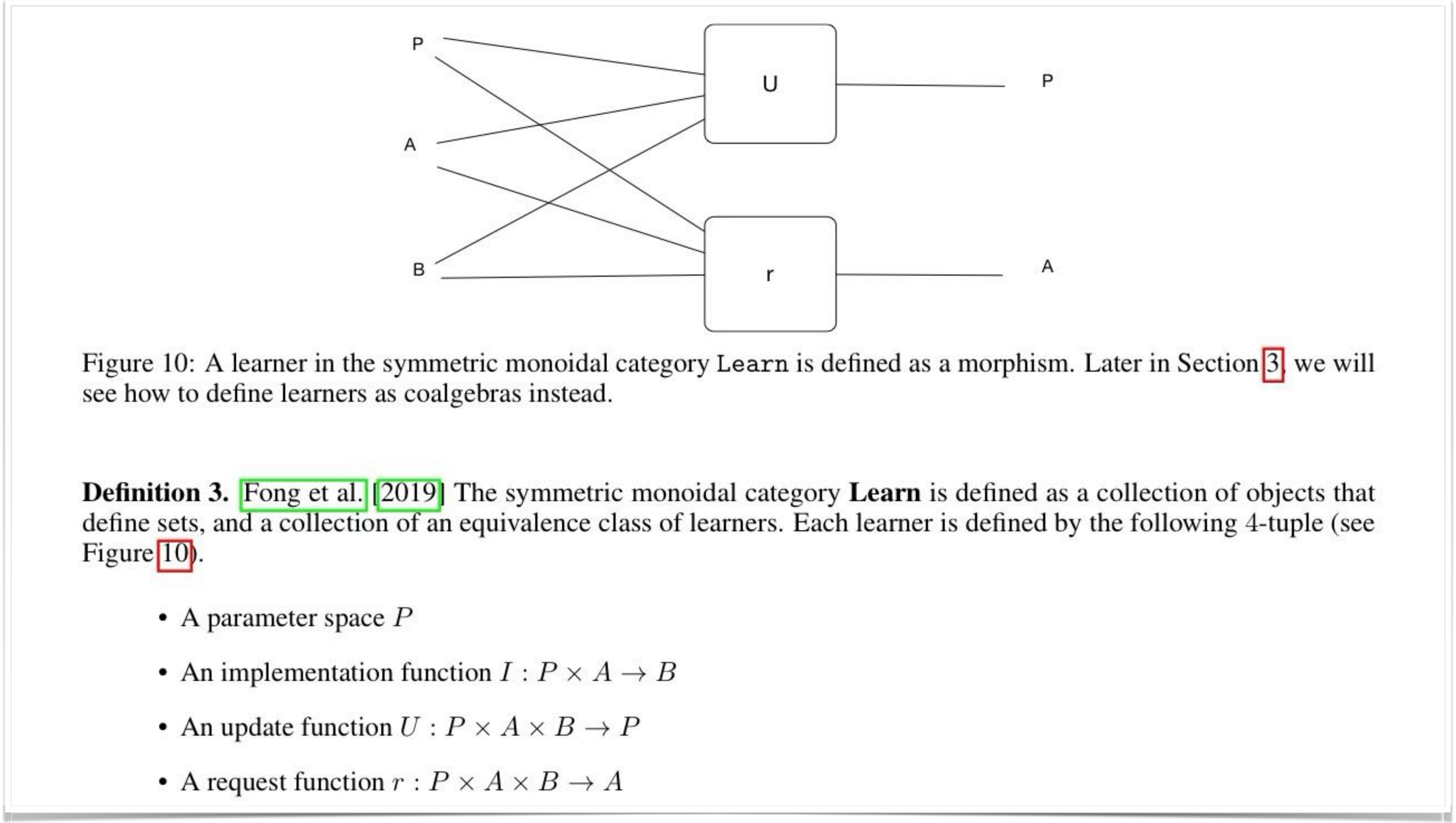
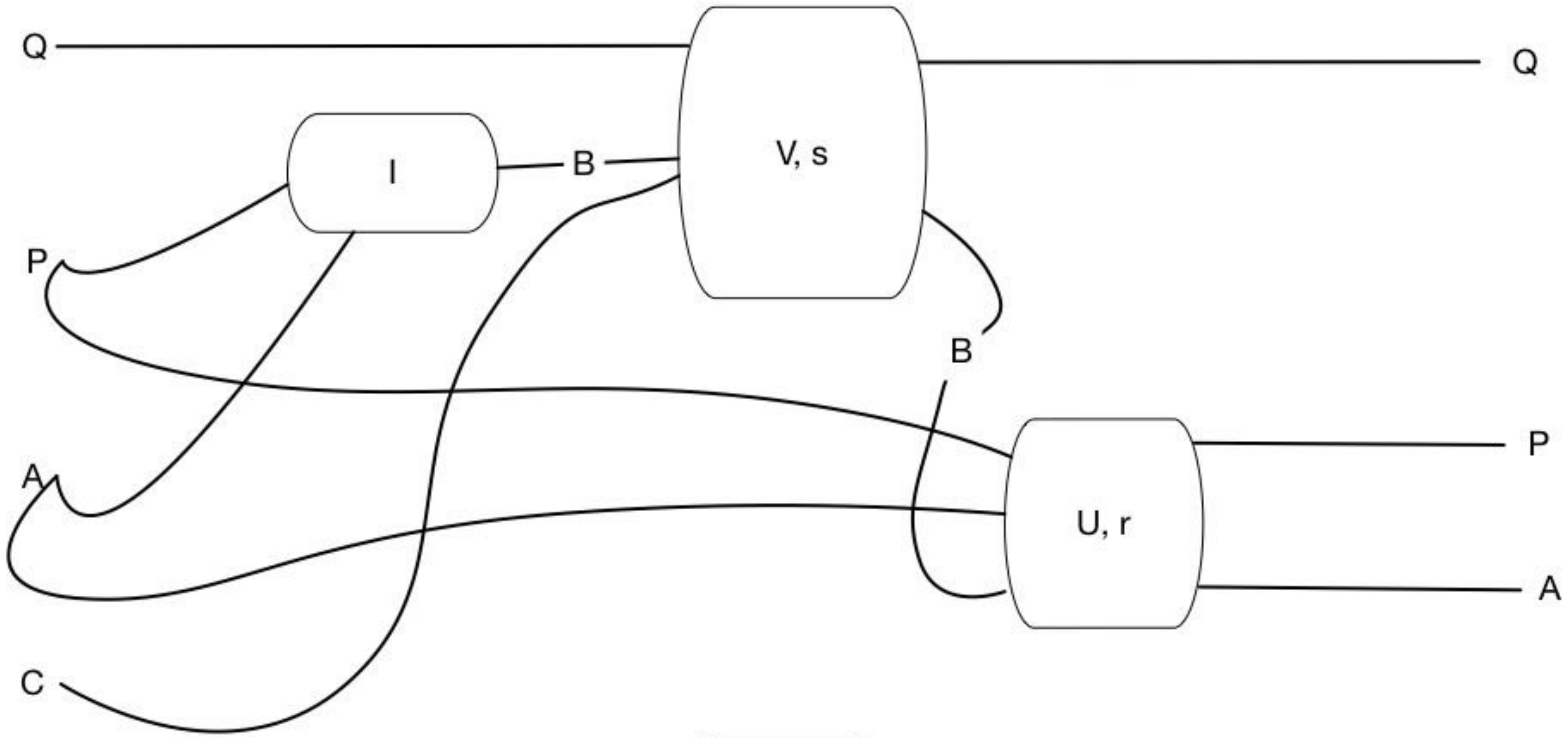


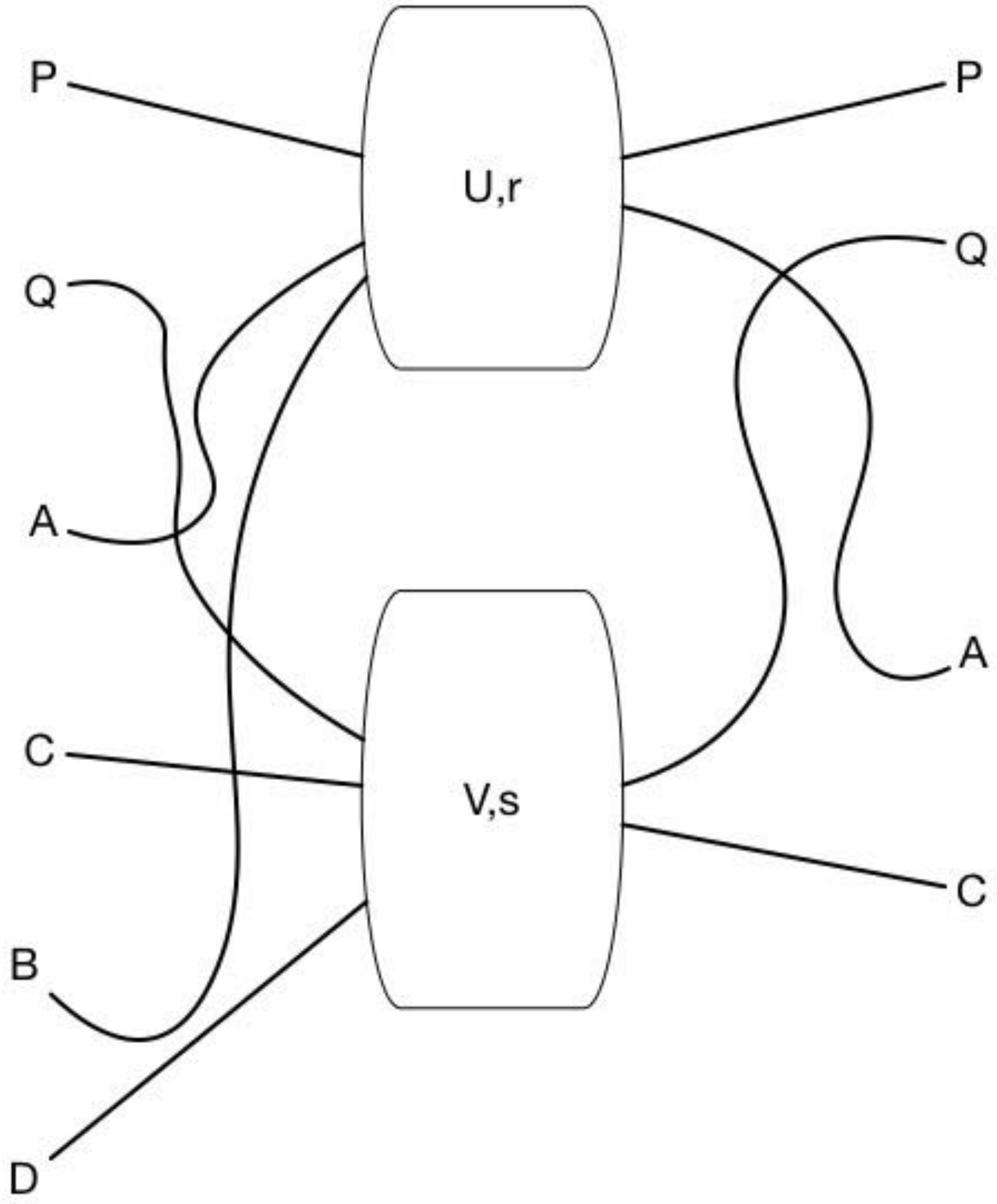
Figure 10: A learner in the symmetric monoidal category `Learn` is defined as a morphism. Later in Section 3, we will see how to define learners as coalgebras instead.

Definition 3. Fong et al. [2019] The symmetric monoidal category **Learn** is defined as a collection of objects that define sets, and a collection of an equivalence class of learners. Each learner is defined by the following 4-tuple (see Figure 10).

- A parameter space P
- An implementation function $I : P \times A \to B$
- An update function $U : P \times A \times B \to P$
- A request function $r : P \times A \times B \to A$



Sequential Composition



Parallel Composition

$$A \xrightarrow{(P,I,U,r)} B \xrightarrow{(Q,J,V,s)} C$$

The composite learner $A \to C$ is defined as $(P \times Q, I \cdot J, U \cdot V, r \cdot s)$, where the composite implementation function is

$$(I \cdot J)(p, q, a) \coloneqq J(q, I(p, a))$$

and the composite update function is

$$U \cdot V(p, q, a, c) \coloneqq (U(p, a, s(q, I(p, a), c))\, , V(q, I(p, a), c)$$

and the composite request function is

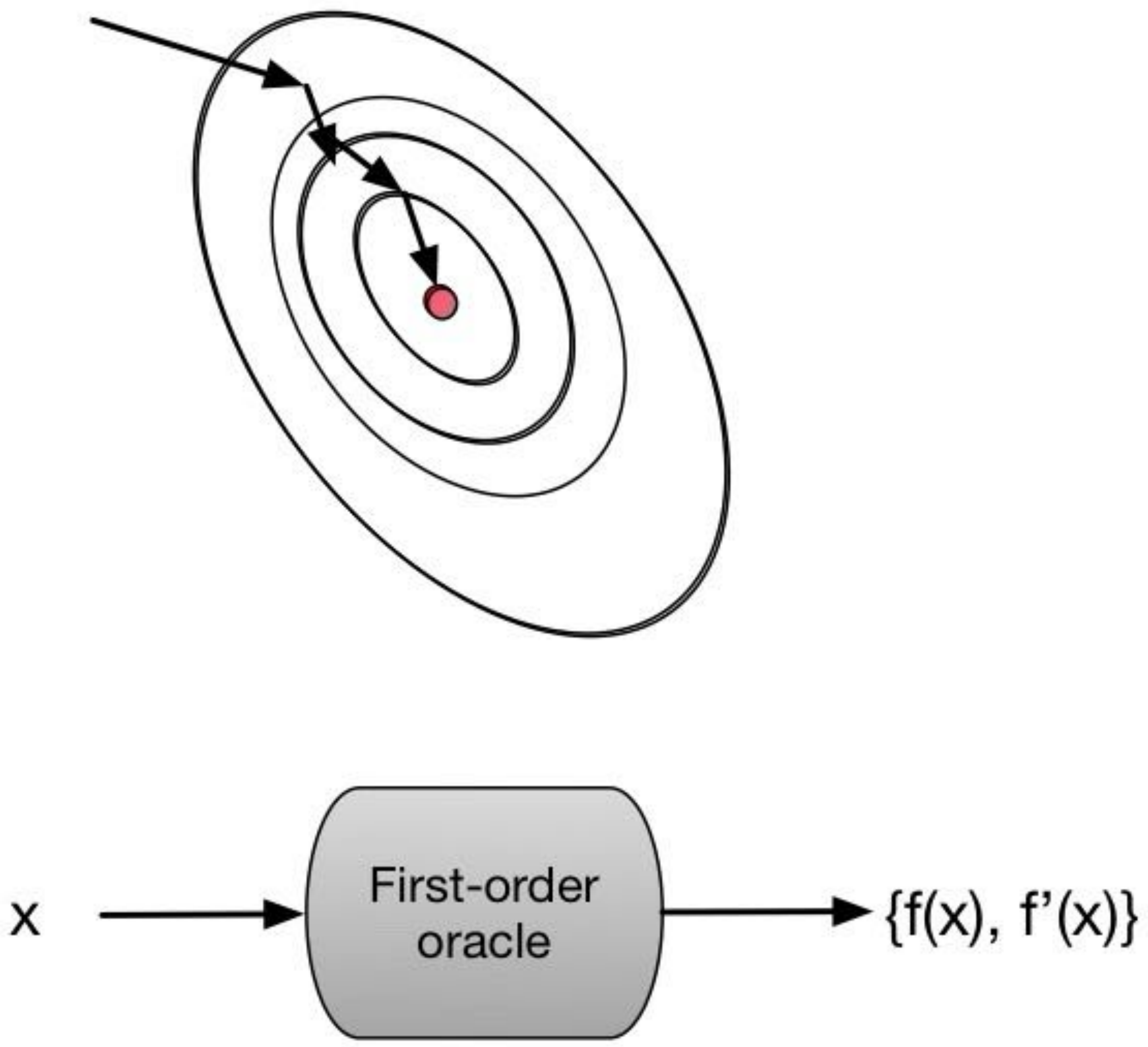
$$(r \cdot s)(p, q, a, c) \coloneqq r(p, a, s(q, I(p, a), c)).$$

$$\begin{tikzcd}
NNet \arrow[rr] \arrow[dr, "F"'] & & Learn \\
& Param \arrow[ur, "L_{\epsilon,e}"] &
\end{tikzcd}$$

Backprop is a functor!

$$U_I(p,a,b) \coloneqq p - \epsilon\nabla_p E_I(p,a,b)$$

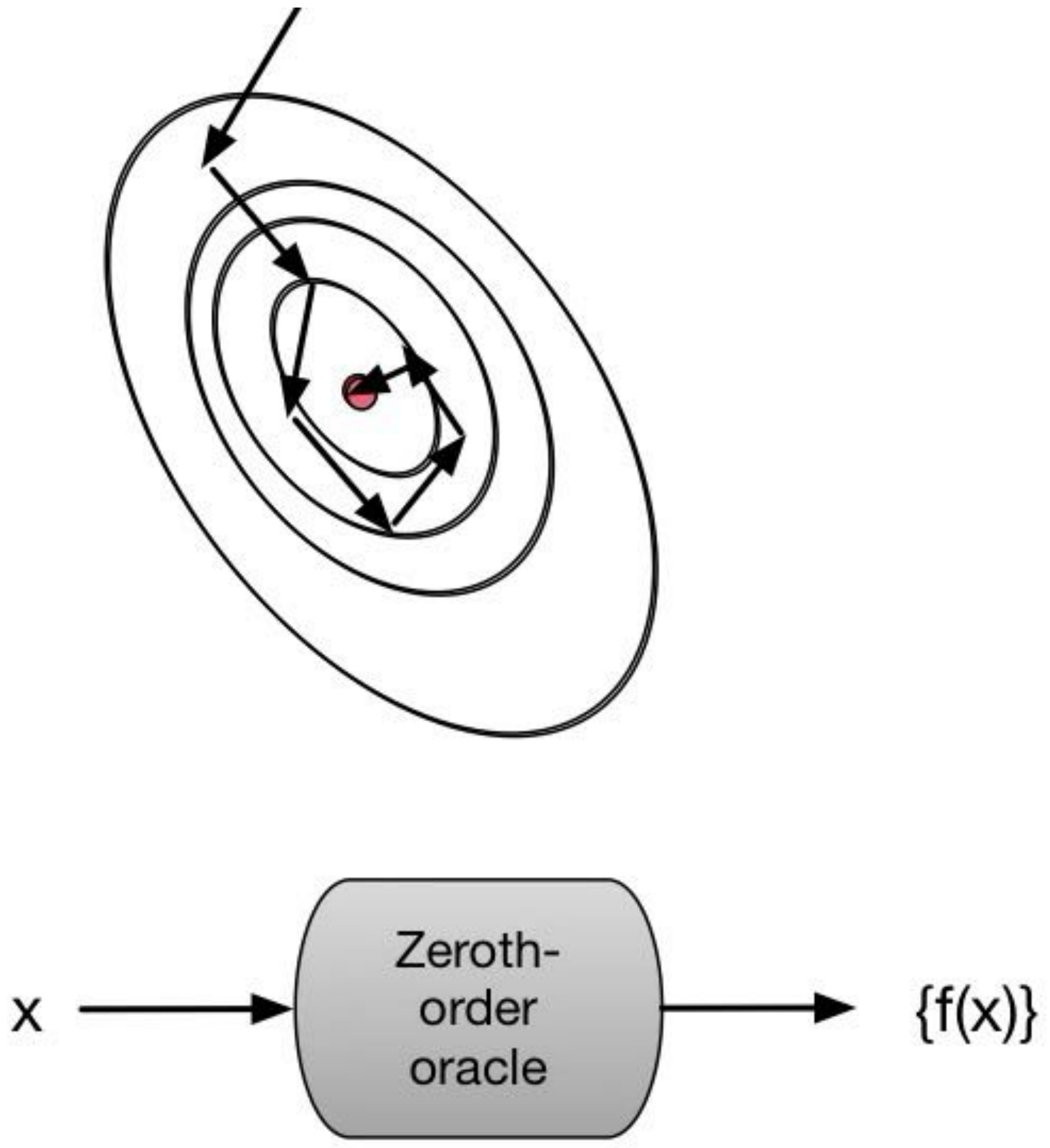
$$r_I(p,a,b) \coloneqq f_a(\nabla_a E_I(p,a,b))$$



x

First-order oracle

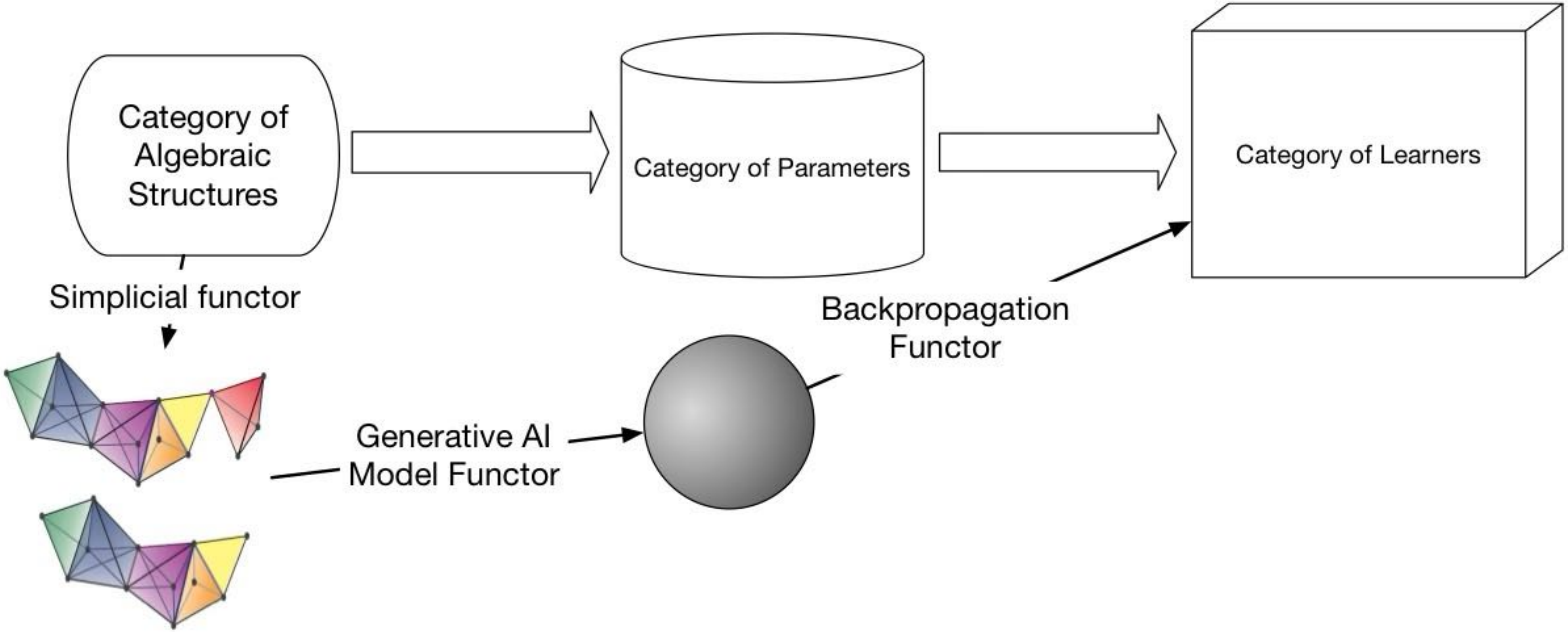
{f(x), f’(x)}



x

Zeroth-order oracle

{f(x)}


Category of Algebraic Structures
Category of Parameters
Category of Learners
Simplicial functor
Backpropagation Functor
Generative AI Model Functor

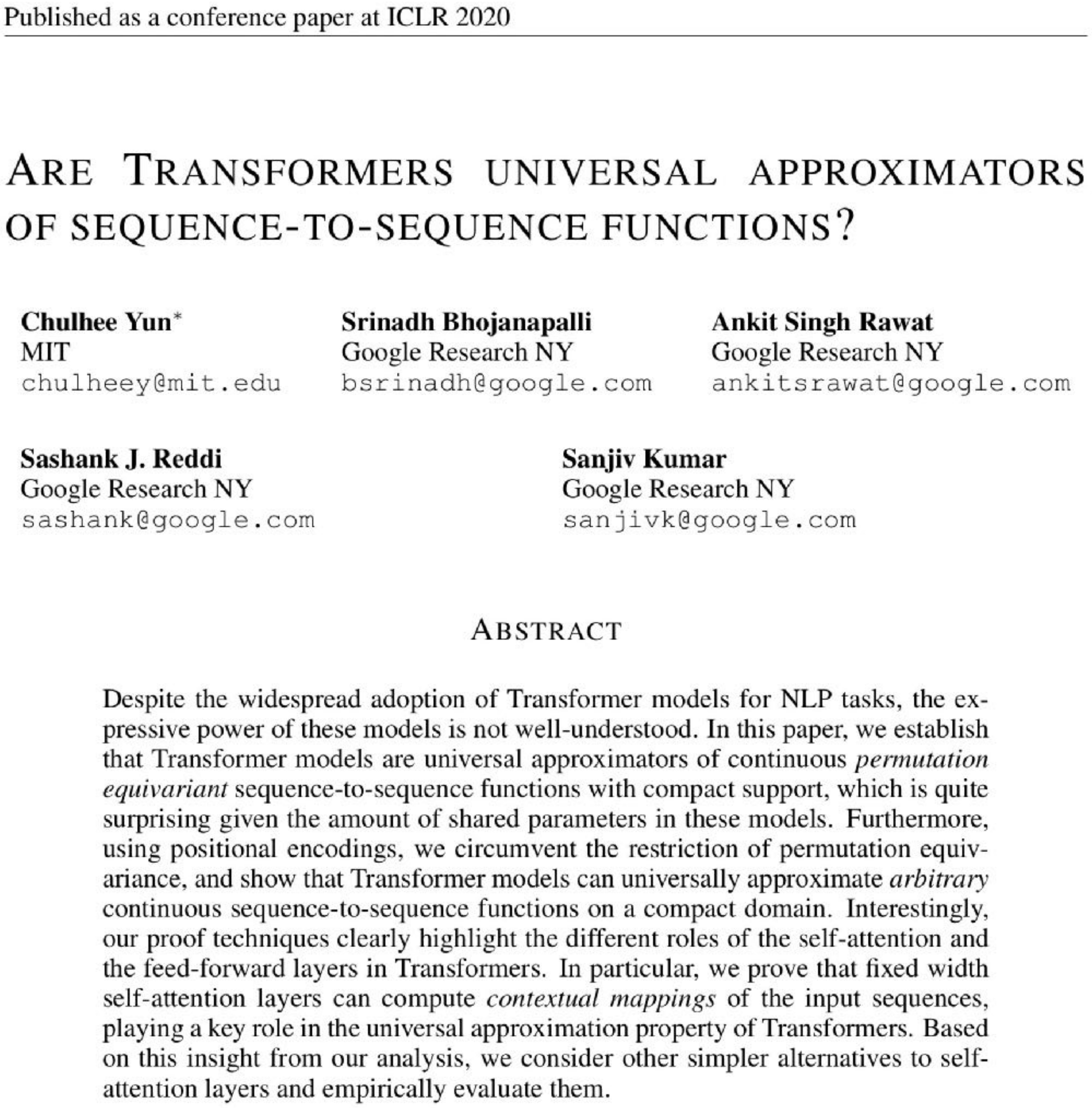
Nerve of a Category

- Recall a category is defined as a collection of objects, and a collection of arrows between any pair of objects

- A simplicial set is a contravariant functor mapping the simplicial category to the category of sets

- Any category can be mapped onto a simplicial set by constructing its nerve

- Intuitively, consider all sequences of composable morphisms of length n!



Published as a conference paper at ICLR 2020

ARE TRANSFORMERS UNIVERSAL APPROXIMATORS OF SEQUENCE-TO-SEQUENCE FUNCTIONS?

Chulhee Yun[*]
MIT
chulheey@mit.edu

Srinadh Bhojanapalli
Google Research NY
bsrinadh@google.com

Ankit Singh Rawat
Google Research NY
ankitsrawat@google.com

Sashank J. Reddi
Google Research NY
sashank@google.com

Sanjiv Kumar
Google Research NY
sanjivk@google.com

ABSTRACT

Despite the widespread adoption of Transformer models for NLP tasks, the expressive power of these models is not well-understood. In this paper, we establish that Transformer models are universal approximators of continuous *permutation equivariant* sequence-to-sequence functions with compact support, which is quite surprising given the amount of shared parameters in these models. Furthermore, using positional encodings, we circumvent the restriction of permutation equivariance, and show that Transformer models can universally approximate *arbitrary* continuous sequence-to-sequence functions on a compact domain. Interestingly, our proof techniques clearly highlight the different roles of the self-attention and the feed-forward layers in Transformers. In particular, we prove that fixed width self-attention layers can compute *contextual mappings* of the input sequences, playing a key role in the universal approximation property of Transformers. Based on this insight from our analysis, we consider other simpler alternatives to self-attention layers and empirically evaluate them.

Permutation-equivariant functions

f(XP) = f(X)P

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "P"'] & Y \arrow[r, "g"] \arrow[d, "P"] & Z \arrow[d, "P"] \\
XP \arrow[r, "f"'] & YP \arrow[r, "g"'] & ZP
\end{tikzcd}$$

$$\begin{aligned}\mathrm{Attn}(X) &= X + \sum_{i=1}^{h} W_O^i W_V^i X \cdot \sigma[W_K^i X)^T W_Q^i X] \\ \mathrm{FF}(X) &= \mathrm{Attn}(X) + W_2 \cdot \mathrm{ReLU}(W_1 \cdot \mathrm{Attn}(X) + b_1 \mathbf{1}_n^T,\end{aligned}$$

Definition 32. The category $\mathcal{C}_T$ of Transformer models is defined as follows:

- The objects Obj(C) are defined as vectors $X \in \mathbb{R}^{d \times n}$ denoting n-length sequences of tokens of dimension d.
- The arrows or morphisms of the category $\mathcal{C}_T$ are defined as a family of sequence-to-sequence functions and defined as:

$$T^{h,m,r} := \{f : \mathbb{R}^{d \times n} \to \mathbb{R}^{d \times n} \mid \text{where } f(XP) = XP, \text{ for some permutation matrix } P\}$$

Nerve of the Category of Transformers

- Since Transformers define a category over Euclidean spaces of permutation-equivariant functions, we can construct its nerve
- Consider all compositions of Transformers building blocks of length n
- This construction maps the category of Transformers into a simplicial set
- It is a full and faithful embedding of Transformers as simplicial sets
- However, simplicial sets cannot be faithfully mapped back to ordinary categories

Simplicial Sets vs. Categories

- Any category can be embedded faithfully into a simplicial set using its nerve
- The embedding is full and faithful (perfect reconstruction)
- Unfortunately, the converse is not possible
- Given a simplicial set, the left adjoint functor that maps it into a category is lossy!
- GAIA (in theory!) is more powerful than existing generative AI formalisms

Summary

- Deep learning faces an energy crisis
- Architectures like Transformers are fundamentally limited!
- We need a better framework: GAIA is one possible approach
- Builds on higher-order category theory of simplicial sets
- GAIA is a **theoretical** framework — not yet an actual working system!