The Algorithmic Weltanschauung: An Algorithmic, Platonic Perspective

Giulio Ruffini

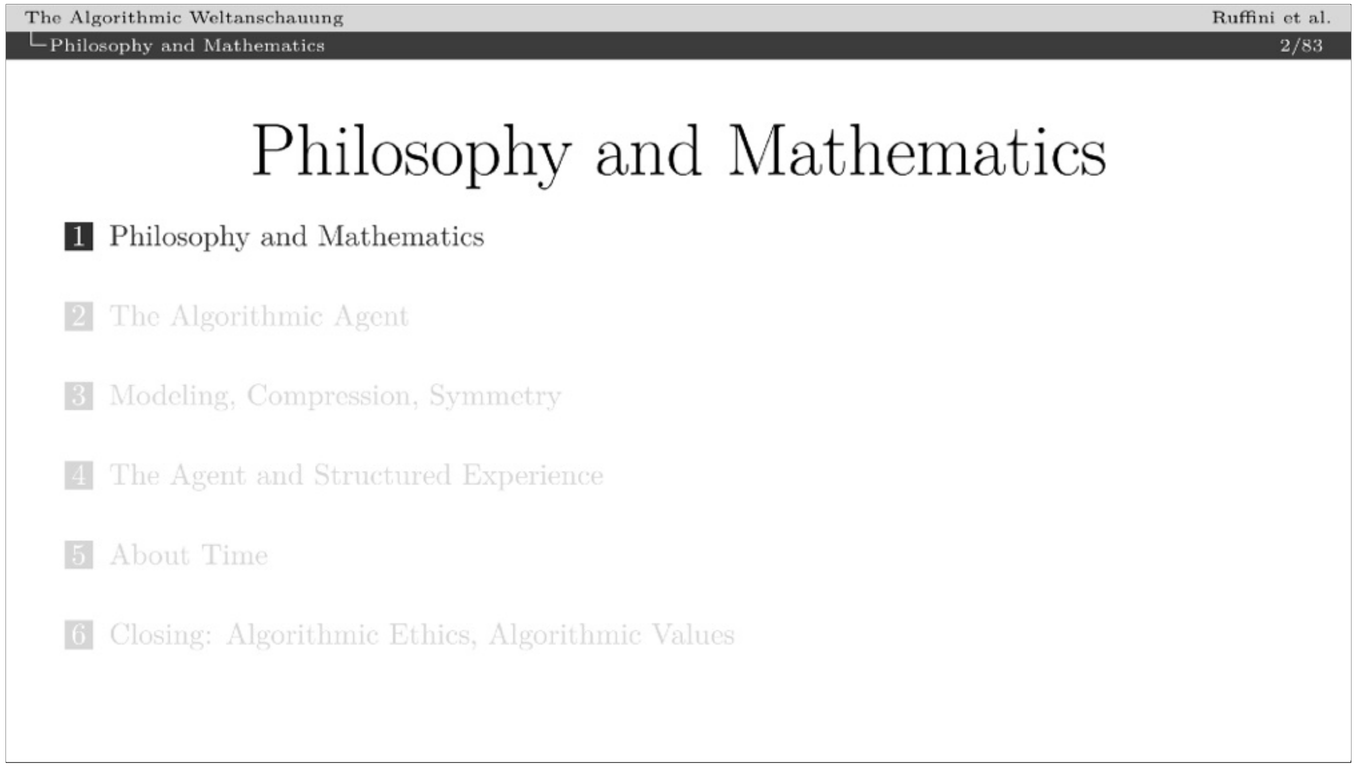
Brain Modeling Department, Neuroelectrics Barcelona, Barcelona
Barcelona Computational Foundation (BCOM.one)
(Platonic series, Dec, 2025)

giulio.ruffini@neuroelectrics.com

Hello. I am Giulio Ruffini from Neuroelectrics Barcelona and the Barcelona Computational Foundation. I am a theoretical physicist and computational neuroscientist working at Neuroelectrics Barcelona (a company dedicated to the creation of brain stimulation solutions in the clinical sector) and co-founder of the Barcelona Computational Foundation (more on this later).

Today, I will be presenting 'The Algorithmic Weltanschauung' or World View—an algorithmic, Platonic perspective on reality, which I think fits well with our Symposium.

Many thanks to Michael for the invitation!



Philosophy and Mathematics

1 Philosophy and Mathematics

2 The Algorithmic Agent

3 Modeling, Compression, Symmetry

4 The Agent and Structured Experience

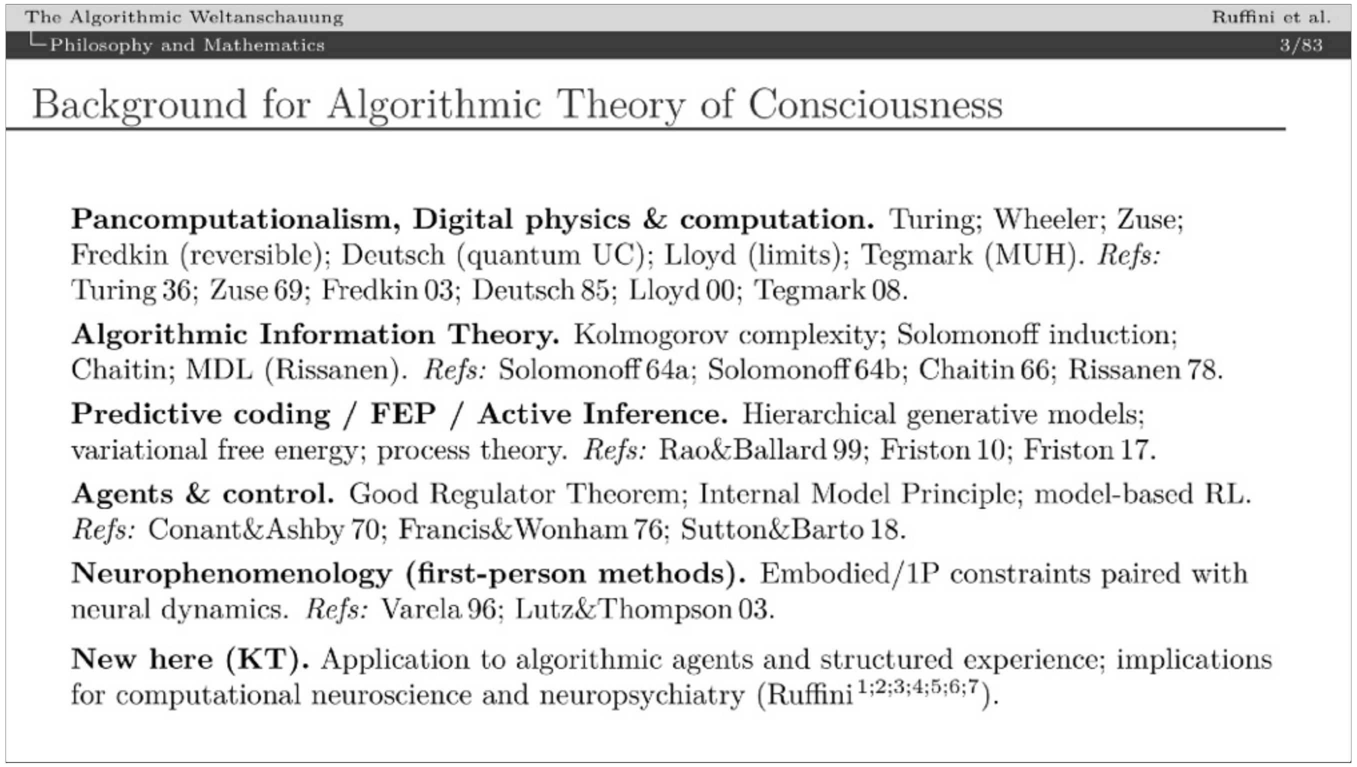
5 About Time

6 Closing: Algorithmic Ethics, Algorithmic Values

Here is the roadmap for the talk. We will begin by bridging Philosophy and Mathematics, then bridge to time and computation to ultimately define the Algorithmic Agent.

We will then explore how the notion of “World Model” corresponds to Compression and Symmetry, leading us to the concept of Structured Experience.

Finally, we will conclude with thoughts on subjective time (*chronoception*) and Algorithmic Ethics.



Background for Algorithmic Theory of Consciousness

Pancomputationalism, Digital physics & computation. Turing; Wheeler; Zuse; Fredkin (reversible); Deutsch (quantum UC); Lloyd (limits); Tegmark (MUH). *Refs:* Turing 36; Zuse 69; Fredkin 03; Deutsch 85; Lloyd 00; Tegmark 08.

Algorithmic Information Theory. Kolmogorov complexity; Solomonoff induction; Chaitin; MDL (Rissanen). *Refs:* Solomonoff 64a; Solomonoff 64b; Chaitin 66; Rissanen 78.

Predictive coding / FEP / Active Inference. Hierarchical generative models; variational free energy; process theory. *Refs:* Rao&Ballard 99; Friston 10; Friston 17.

Agents & control. Good Regulator Theorem; Internal Model Principle; model-based RL. *Refs:* Conant&Ashby 70; Francis&Wonham 76; Sutton&Barto 18.

Neurophenomenology (first-person methods). Embodied/1P constraints paired with neural dynamics. *Refs:* Varela 96; Lutz&Thompson 03.

New here (KT). Application to algorithmic agents and structured experience; implications for computational neuroscience and neuropsychiatry (Ruffini[1;2;3;4;5;6;7]).

This work stands on the shoulders of quite a few giants. We build upon Pancomputationalism and digital physics—from Llull, Al-karizmi, to Turing, Zuse, Chaitin, Solmonoff, Kolmogorov, Lloyd, Bennet, Fredkin, Deutsch, and Tegmark (to name a few).

We will rely heavily on Algorithmic Information Theory (Kolmogorov, Solomonoff, Chaitin) and Active Inference frameworks like Karl Friston's.

Our goal is to build on this to define the notion of algorithmic agent and, around them, a science of structured experience.

For references on my own work, please visit giulioruffini.github.io (in particular https://giulioruffini.github.io/kt/).

Questions

What is Reality?
What is consciousness?
Why is math so powerful?
What is Time?
And who is this famous observer?
Who am AI?

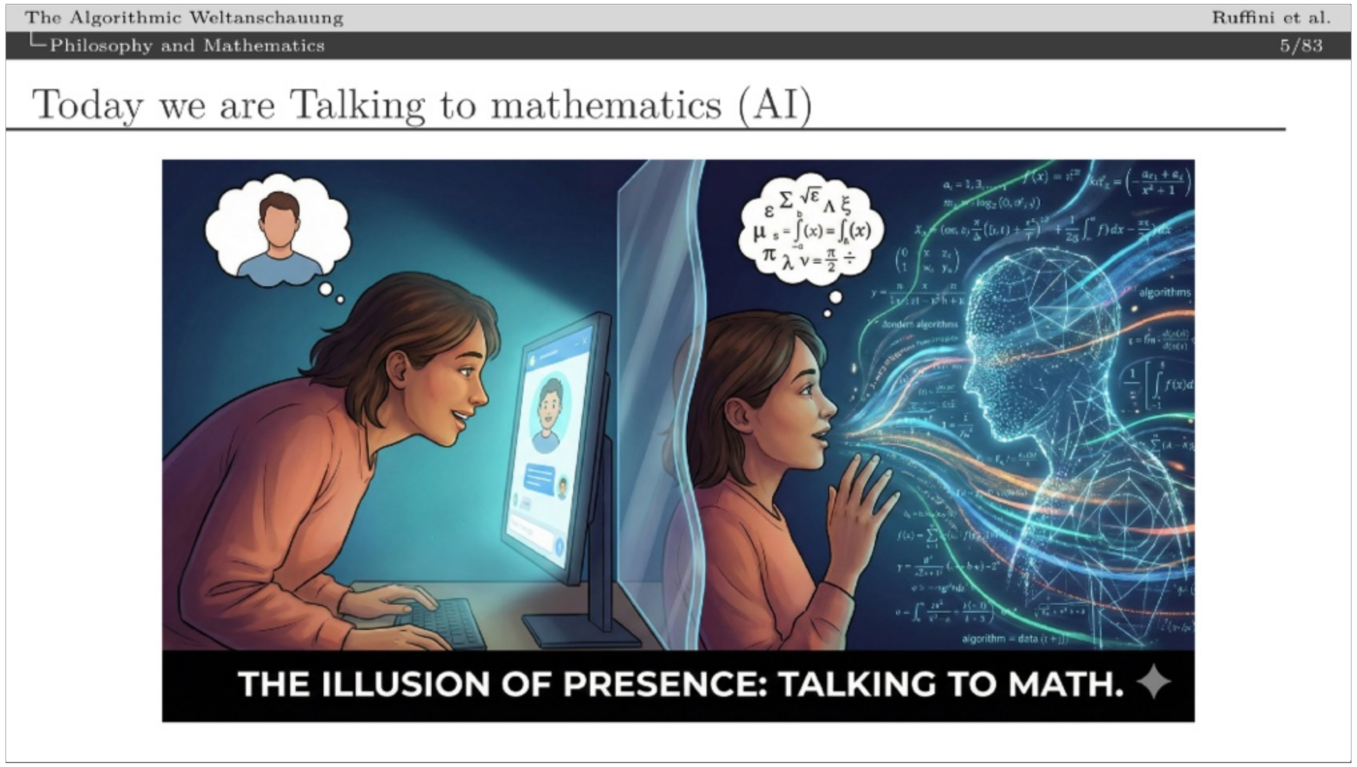
We begin with the fundamental questions that have fascinated thinkers since the dawn of time: What is Reality? What is Consciousness? What is Time?

For centuries, these were the domain of philosophers. But as we enter the age of artificial agents, we are forced to confront them head-on. We must also ask: Who is this 'Observer' that physics relies upon? And who—or what—is the AI looking back at us? “Who am AI?”

It seems like the best view of what reality is about must start at the Observer.

And we must not forget the puzzle posed by Eugene Wigner: the 'unreasonable effectiveness of mathematics.' Why does the physical world adhere so strictly to mathematical laws?

A successful framework must address all these questions.



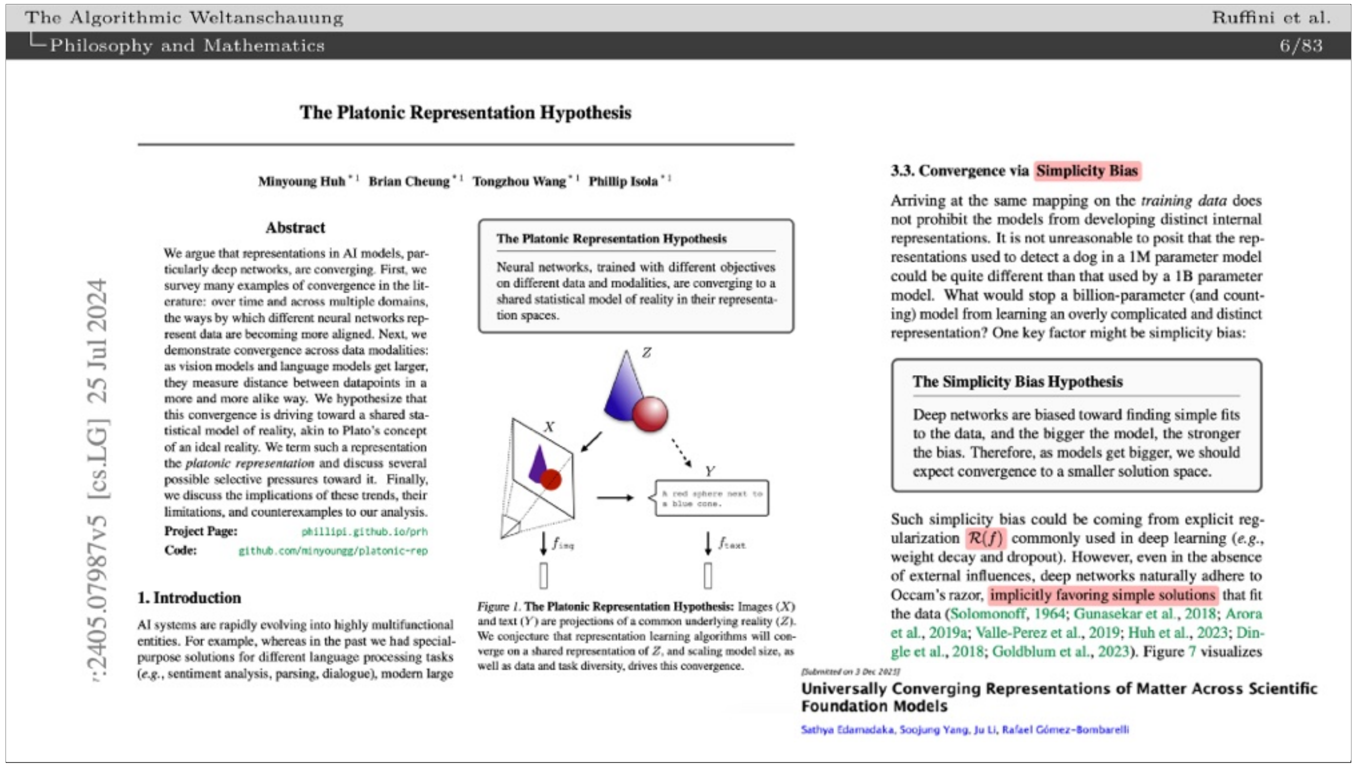
Today we are Talking to mathematics (AI)

THE ILLUSION OF PRESENCE: TALKING TO MATH.

These questions are very timely. When we interact with modern AI systems, we often feel a 'presence', and this perception will only increase with time as AI evolves.

But we must realize we are literally talking to mathematics! The 'illusion of presence' is generated by complex algorithms and data processing. And it is not so different than when we talk to each other, as both machines and humans are organized ensembles.

Mathematics must be at the “bottom” of all this.



The Platonic Representation Hypothesis

Minyoung Huh *1 Brian Cheung *1 Tongzhou Wang *1 Phillip Isola *1

Abstract

We argue that representations in AI models, particularly deep networks, are converging. First, we survey many examples of convergence in the literature: over time and across multiple domains, the ways by which different neural networks represent data are becoming more aligned. Next, we demonstrate convergence across data modalities: as vision models and language models get larger, they measure distance between datapoints in a more and more alike way. We hypothesize that this convergence is driving toward a shared statistical model of reality, akin to Plato's concept of an ideal reality. We term such a representation the *platonic representation* and discuss several possible selective pressures toward it. Finally, we discuss the implications of these trends, their limitations, and counterexamples to our analysis.

Project Page: phillipi.github.io/prh
Code: github.com/minyoungg/platonic-rep

1. Introduction

AI systems are rapidly evolving into highly multifunctional entities. For example, whereas in the past we had special-purpose solutions for different language processing tasks (*e.g.*, sentiment analysis, parsing, dialogue), modern large

The Platonic Representation Hypothesis

Neural networks, trained with different objectives on different data and modalities, are converging to a shared statistical model of reality in their representation spaces.

Figure 1. **The Platonic Representation Hypothesis:** Images (X) and text (Y) are projections of a common underlying reality (Z). We conjecture that representation learning algorithms will converge on a shared representation of Z, and scaling model size, as well as data and task diversity, drives this convergence.

3.3. Convergence via Simplicity Bias

Arriving at the same mapping on the *training data* does not prohibit the models from developing distinct internal representations. It is not unreasonable to posit that the representations used to detect a dog in a 1M parameter model could be quite different than that used by a 1B parameter model. What would stop a billion-parameter (and counting) model from learning an overly complicated and distinct representation? One key factor might be simplicity bias:

The Simplicity Bias Hypothesis

Deep networks are biased toward finding simple fits to the data, and the bigger the model, the stronger the bias. Therefore, as models get bigger, we should expect convergence to a smaller solution space.

Such simplicity bias could be coming from explicit regularization $\mathcal{R}(f)$ commonly used in deep learning (*e.g.*, weight decay and dropout). However, even in the absence of external influences, deep networks naturally adhere to Occam's razor, implicitly favoring simple solutions that fit the data (Solomonoff, 1964; Gunasekar et al., 2018; Arora et al., 2019a; Valle-Perez et al., 2019; Huh et al., 2023; Dingle et al., 2018; Goldblum et al., 2023). Figure 7 visualizes

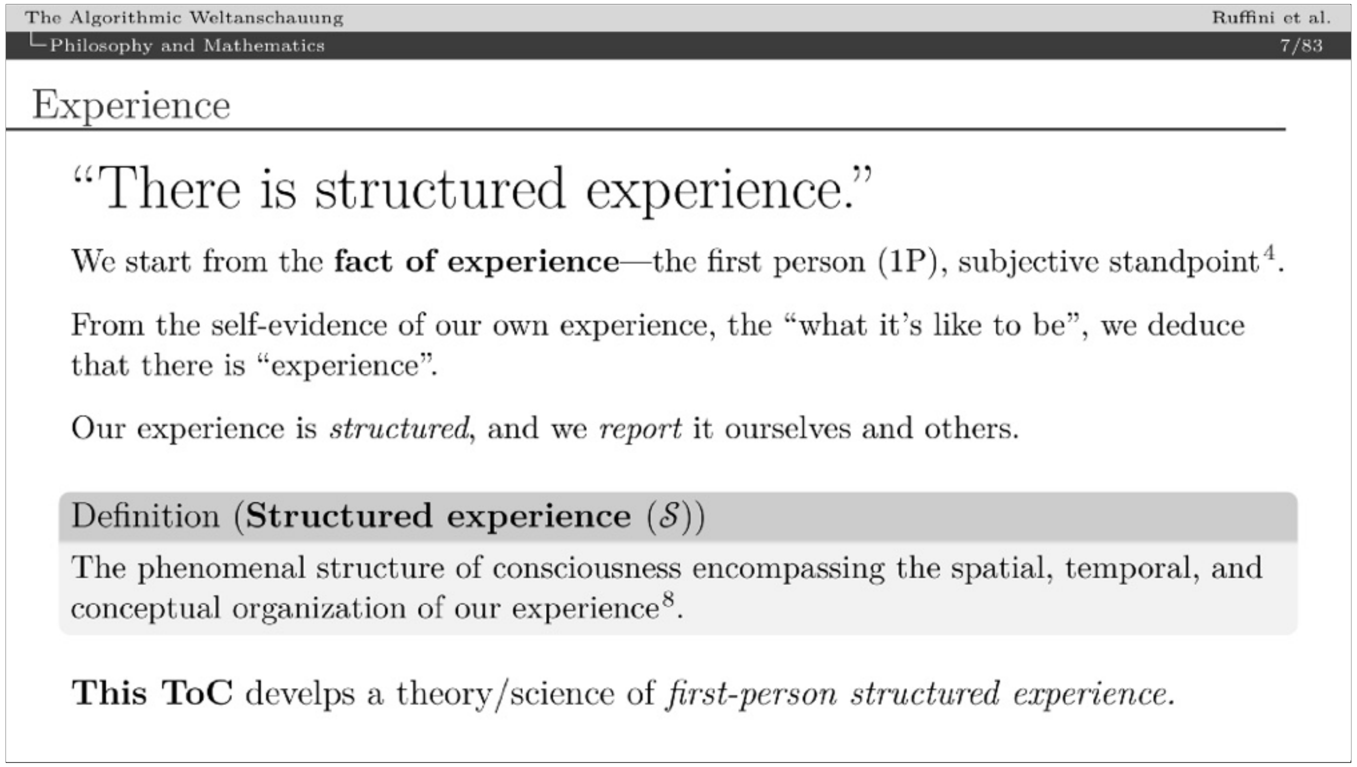
[Submitted on 3 Dec 2023]

Universally Converging Representations of Matter Across Scientific Foundation Models

Sathya Edamadaka, Soojung Yang, Ju Li, Rafael Gómez-Bombarelli

Another recent and fascinating trend in AI is known as the Platonic Representation Hypothesis. Research shows that representations in AI models are converging toward a shared model of reality with embedded *Platonic* forms.

Why should this be? Both machines and humans are capturing the same hidden structure in the world — Platonic forms. This is because of an explicit or implicit 'simplicity bias' in AI systems—where models naturally adhere to Occam's razor, finding simple fits to the data (avoiding “overfitting”), which drives them toward a common 'Platonic' reality that seems to be out there.



Experience

“There is structured experience.”

We start from the **fact of experience**—the first person (1P), subjective standpoint[4].

From the self-evidence of our own experience, the “what it’s like to be”, we deduce that there is “experience”.

Our experience is *structured*, and we *report* it ourselves and others.

Definition (**Structured experience** ($\mathcal{S}$))

The phenomenal structure of consciousness encompassing the spatial, temporal, and conceptual organization of our experience[8].

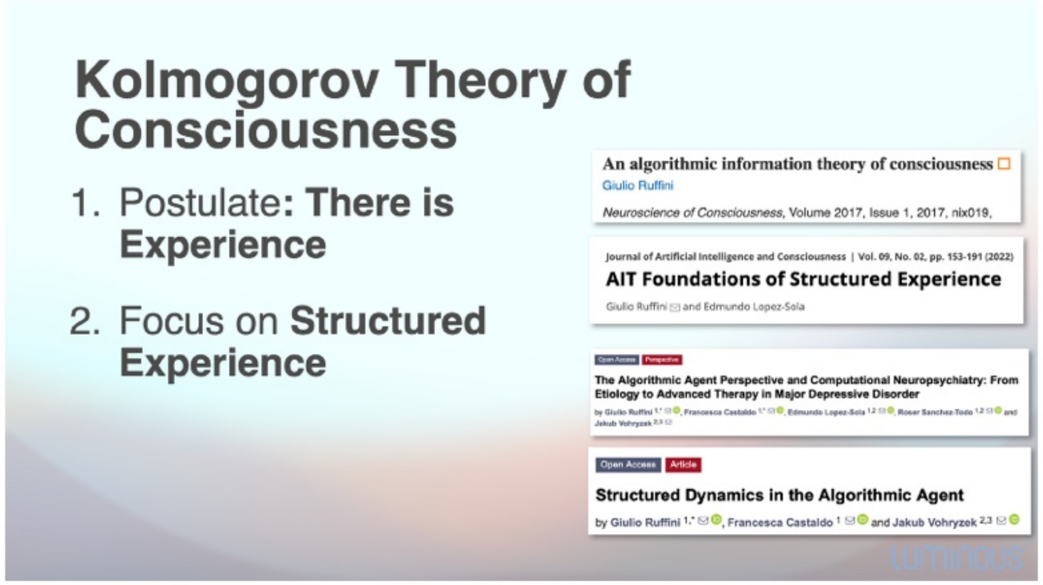
This ToC develps a theory/science of *first-person structured experience.*

To attempt to answer the deep questions, we will pivot to the 'internal' or subjective view.

We must start with the undeniable fact of the first-person perspective, the “what it is like to be”. From this self-evident fact (the only one we can be sure of!), we deduce first that ‘there is experience'.

But not just any experience—our experience is *structured* spatially, temporally, and conceptually.
We define Structured Experience as our phenomenal structure of experience encompassing spatial, temporal, and conceptual organization.

AIT or Kolmogorov Theory of Consciousness



Kolmogorov Theory of Consciousness

1. Postulate**: There is Experience**
2. Focus on **Structured Experience**

An algorithmic information theory of consciousness
Giulio Ruffini
Neuroscience of Consciousness, Volume 2017, Issue 1, 2017, nix019,

Journal of Artificial Intelligence and Consciousness | Vol. 09, No. 02, pp. 153-191 (2022)
AIT Foundations of Structured Experience
Giulio Ruffini and Edmundo Lopez-Sola

Open Access Perspective
The Algorithmic Agent Perspective and Computational Neuropsychiatry: From Etiology to Advanced Therapy in Major Depressive Disorder
by Giulio Ruffini 1,*, Francesca Castaldo 1,*, Edmundo Lopez-Sola 1,2, Roser Sanchez-Todo 1,2 and Jakub Vohryzek 2,3

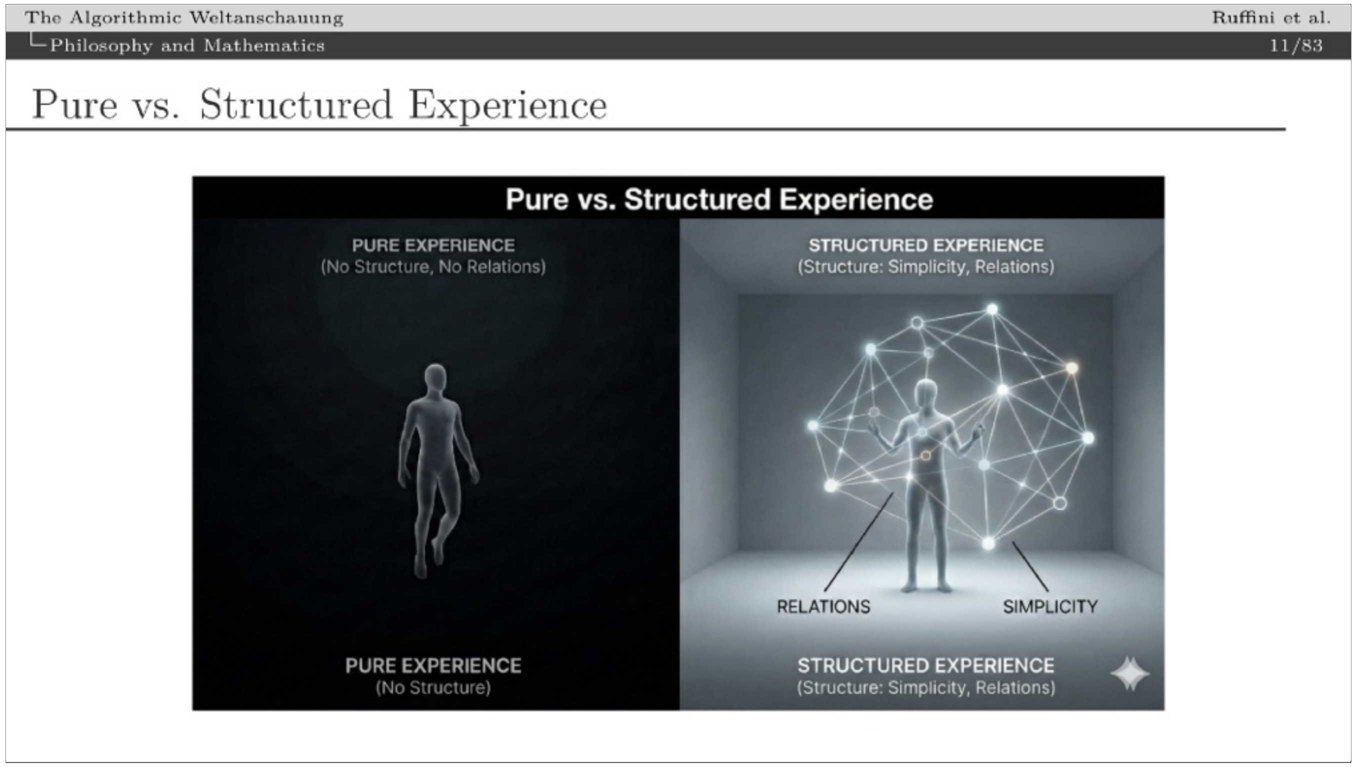
Open Access Article
Structured Dynamics in the Algorithmic Agent
by Giulio Ruffini 1,*, Francesca Castaldo 1 and Jakub Vohryzek 2,3

This is the starting point of the ‘Algorithmic or Kolmogorov Theory of consciousness': *there is structured experience*.

It has been developed during the last 25 years, building on the foundations of Algorithmic Information Theory to applications in computational neuropsychiatry.

It starts from an axiom (there is pure experience), and asks what gives rise to structured experience.

What is indeed “structure”? Well, structure is what mathematics is all about: mathematics is the science of structure.



Pure vs. Structured Experience

Pure vs. Structured Experience
PURE EXPERIENCE
(No Structure, No Relations)
STRUCTURED EXPERIENCE
(Structure: Simplicity, Relations)
RELATIONS
SIMPLICITY
PURE EXPERIENCE
(No Structure)
STRUCTURED EXPERIENCE
(Structure: Simplicity, Relations)

We must distinguish between two states. One (on the left) is 'Pure Experience'—a raw, structureless state with no relations, providing the experience substrate.

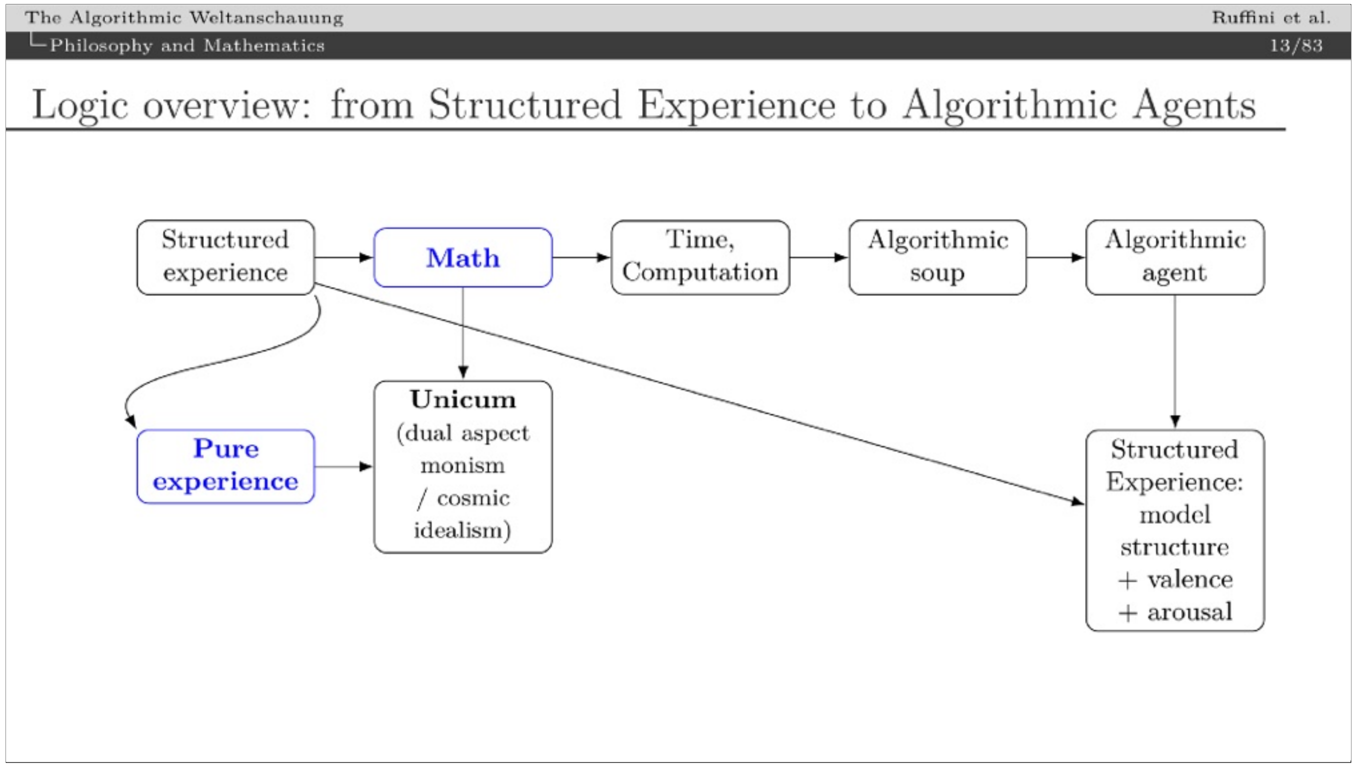
On the right, we portray 'Structured Experience'—defined by experience with relations, structure and simplicity. This is the state of being we actually inhabit, but we can infer the existence of the former through reason. For example, we can define pure experience as the intersection of all possible experiences. What remains in the intersection is the raw common substrate– *pure* or *primordial* experience.

I THINK, THEREFORE I HAVE STRUCTURED EXPERIENCE!
ADVANCED THINKING FOR DUMMIES

Descartes famously said, 'I think, therefore I am.' We update this to state first: '*I think, therefore I have structured experience*'.

THERE IS STRUCTURED EXPERIENCE. THEREFORE THERE IS (PURE) EXPERIENCE AND MATHEMATICS
ADVANCED THINKING FOR DUMMIES

And from the existence of structured experience, we can deduce the existence of the raw material—pure experience—and the rules that organize it—mathematics.



Logic overview: from Structured Experience to Algorithmic Agents

Structured experience

Math

Time, Computation

Algorithmic soup

Algorithmic agent

Pure experience

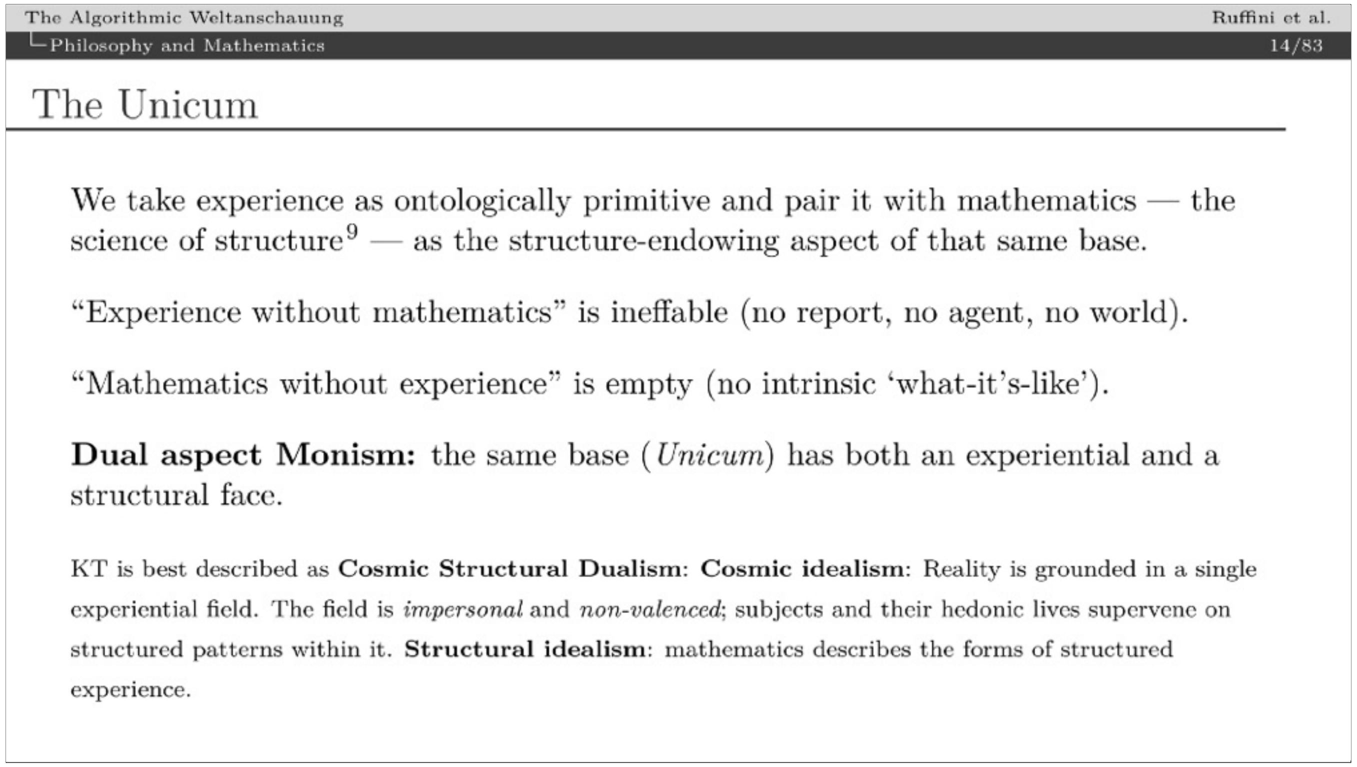
Unicum (dual aspect monism / cosmic idealism)

Structured Experience: model structure + valence + arousal

Here is the logical skeleton of our theory, which we will develop in the talk.

We started with Structured Experience to deduce the key pillars of reality: Experience and Mathematics (the science of structure!). From this mathematical tiling, we seek to unearth the origins of Time and Computation, which can then be used to build the 'Algorithmic Soup' in which Algorithmic Agents such as us arise.

Ultimately, reality stems from the 'Unicum'—a dual-aspect monism consisting of Experience and Mathematics.



The Unicum

We take experience as ontologically primitive and pair it with mathematics — the science of structure[9] — as the structure-endowing aspect of that same base.

“Experience without mathematics” is ineffable (no report, no agent, no world).

“Mathematics without experience” is empty (no intrinsic ‘what-it’s-like’).

Dual aspect Monism: the same base (*Unicum*) has both an experiential and a structural face.

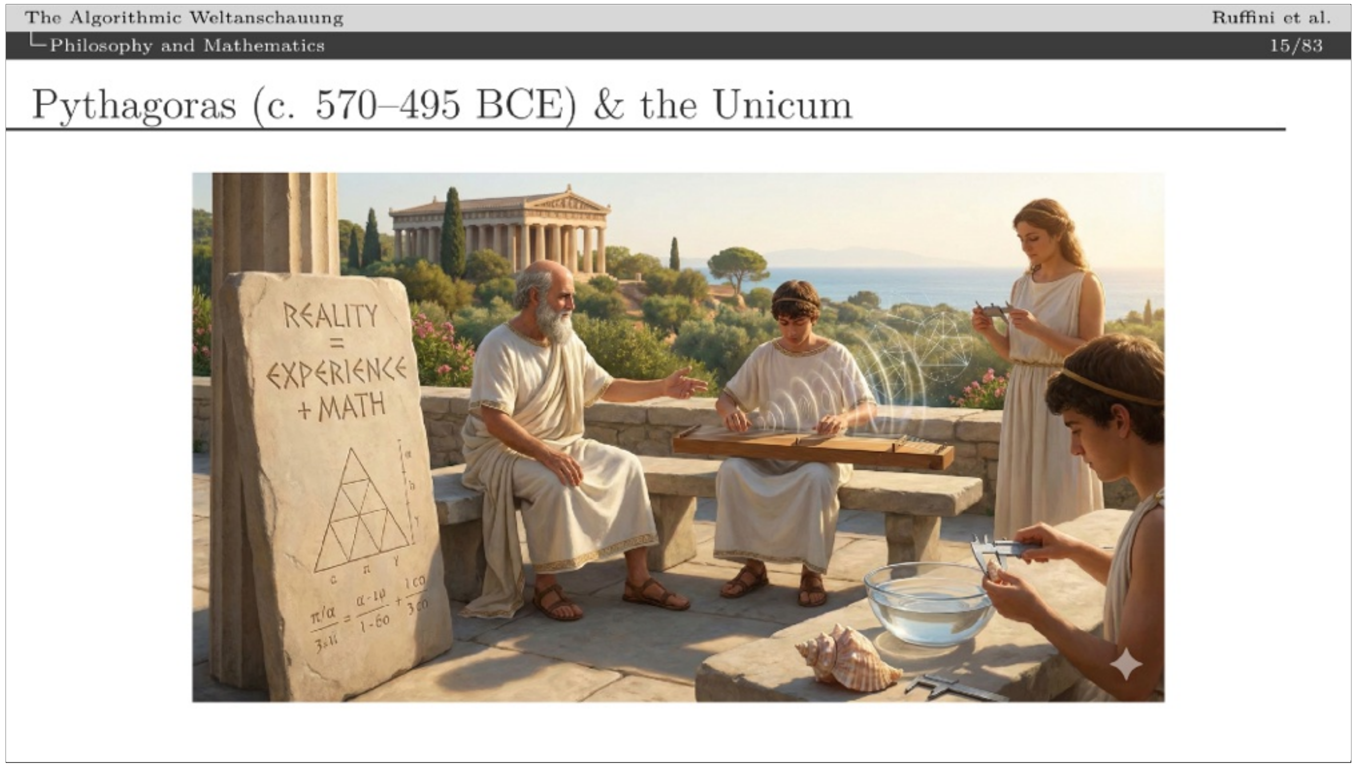
KT is best described as **Cosmic Structural Dualism**: **Cosmic idealism**: Reality is grounded in a single experiential field. The field is *impersonal* and *non-valenced*; subjects and their hedonic lives supervene on structured patterns within it. **Structural idealism**: mathematics describes the forms of structured experience.

The Unicum is the base of reality. Experience is the ontological primitive, and mathematics is its structure-endowing aspect.

They are both needed (necessary ingredients):

Experience without math is ineffable; mathematics without experience is empty.

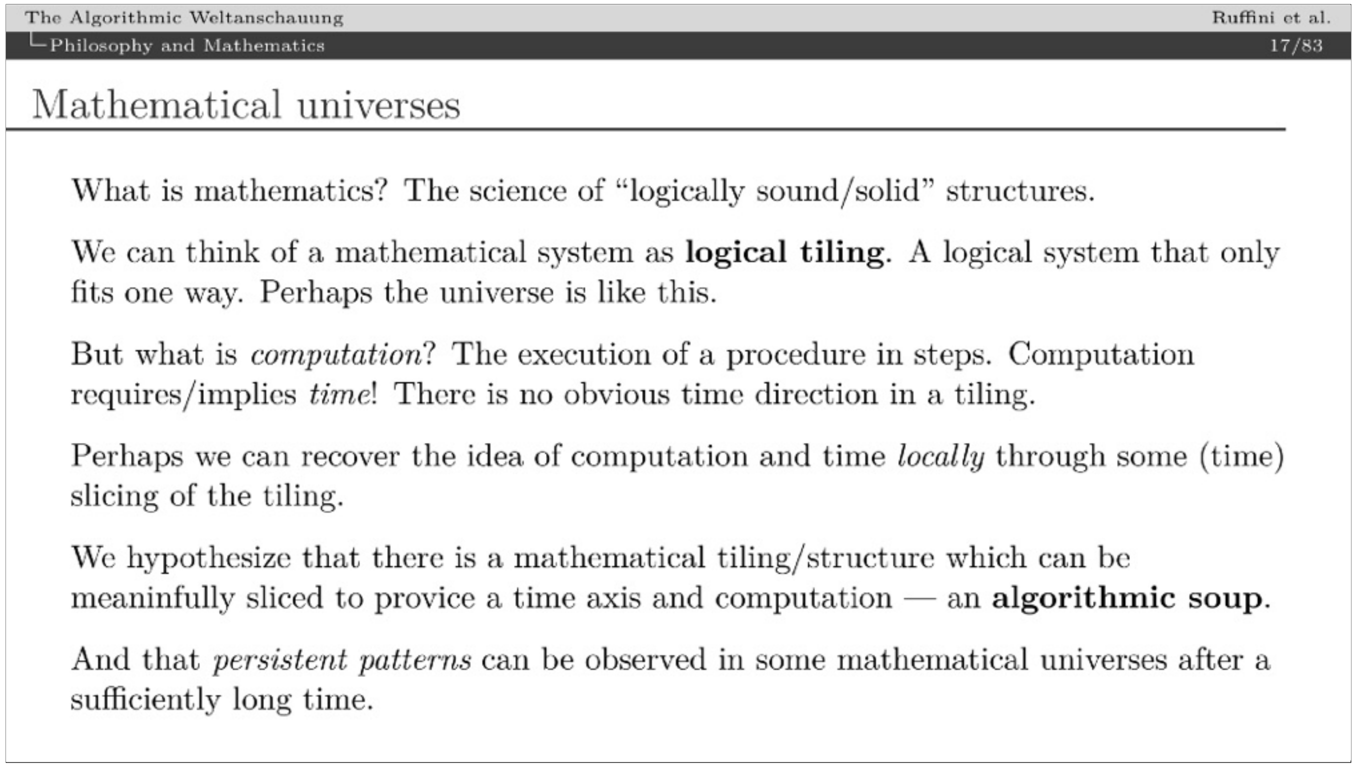
This worldview is a form of Structural Idealism.



Pythagoras (c. 570–495 BCE) & the Unicum

In a way, this view is very aligned with Pythagoras and Plato.

Reality equals Experience plus Mathematics (number in the language of the time). And structured experience arises from mathematical forms.



Mathematical universes

What is mathematics? The science of “logically sound/solid” structures.

We can think of a mathematical system as **logical tiling**. A logical system that only fits one way. Perhaps the universe is like this.

But what is *computation*? The execution of a procedure in steps. Computation requires/implies *time*! There is no obvious time direction in a tiling.

Perhaps we can recover the idea of computation and time *locally* through some (time) slicing of the tiling.

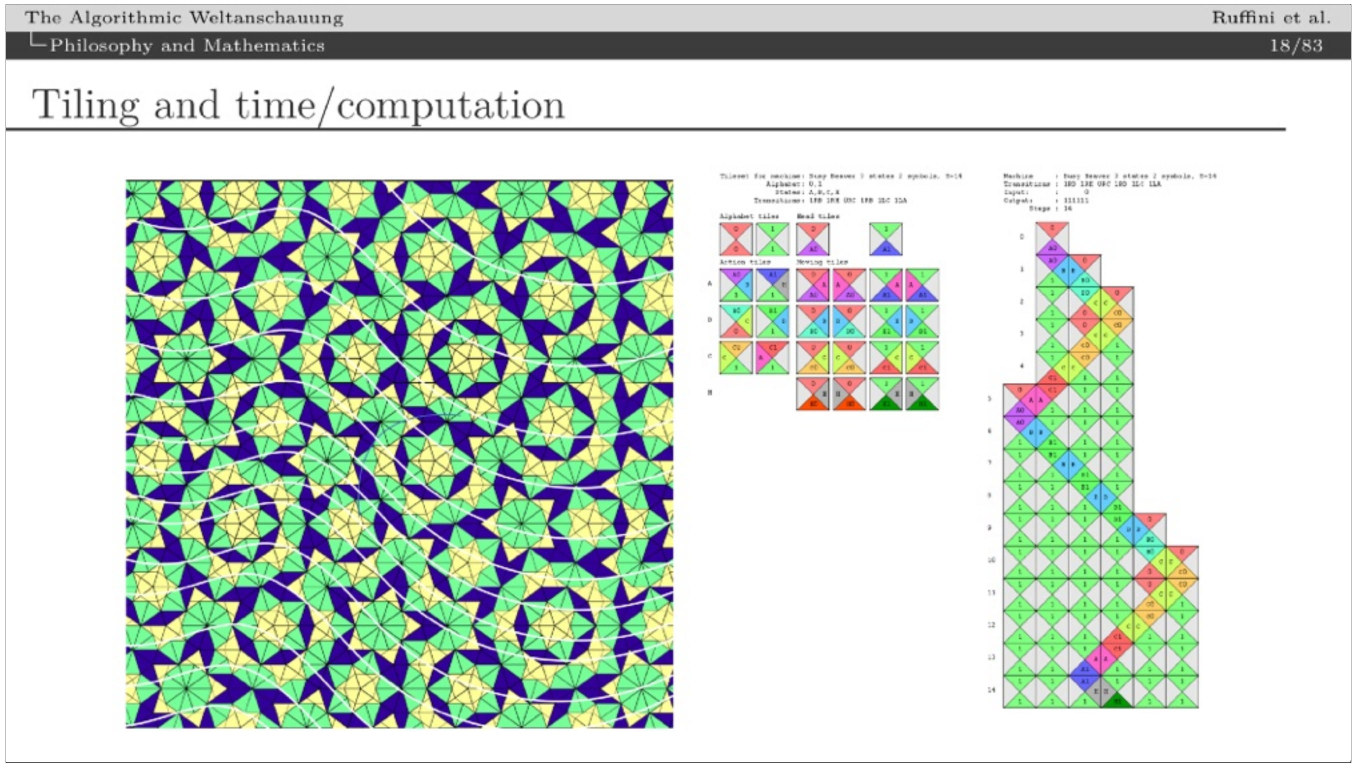
We hypothesize that there is a mathematical tiling/structure which can be meaninfully sliced to provice a time axis and computation — an **algorithmic soup**.

And that *persistent patterns* can be observed in some mathematical universes after a sufficiently long time.

Mathematics is the science of structure—a logical lattice built from axioms and theorems. If we view reality as deriving from such a structure, we might conceptualize it as a 'logical tiling' that fits together perfectly. However, this model presents a paradox: a logical graph is static, containing no intrinsic notion of time or causality.

This contrasts with our experience as agents. We operate through computation (modeled by Turing machines), which necessitates time—a linear dimension of sequential steps. How do we reconcile a timeless structure with temporal process?

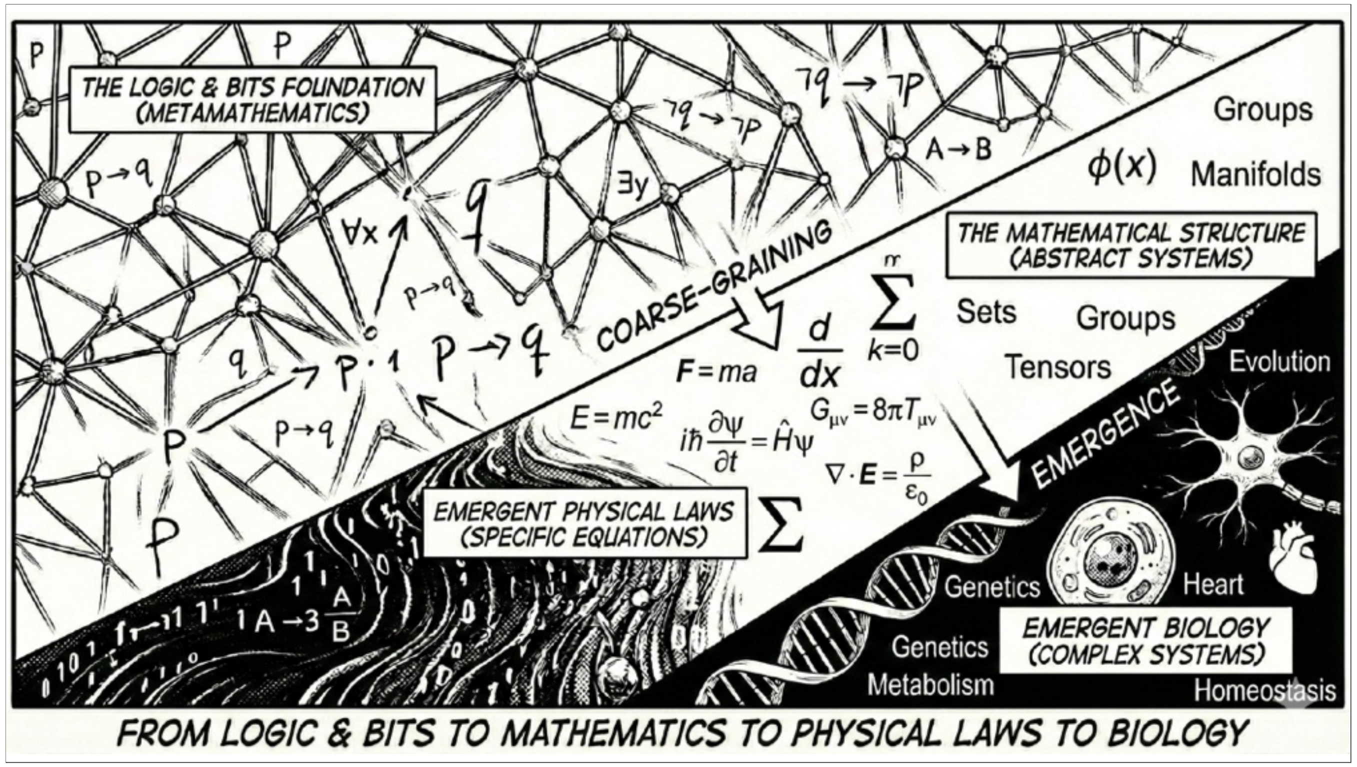
We propose that time and computation arise only when we 'slice' this static tiling in a specific direction. If a valid slicing exists, it allows us to infer the state of one slice from its neighbor, effectively turning static geometry into dynamic derivation. This emergence of sequential rules creates an 'algorithmic soup,' an environment where computational patterns can execute, compete, and persist.



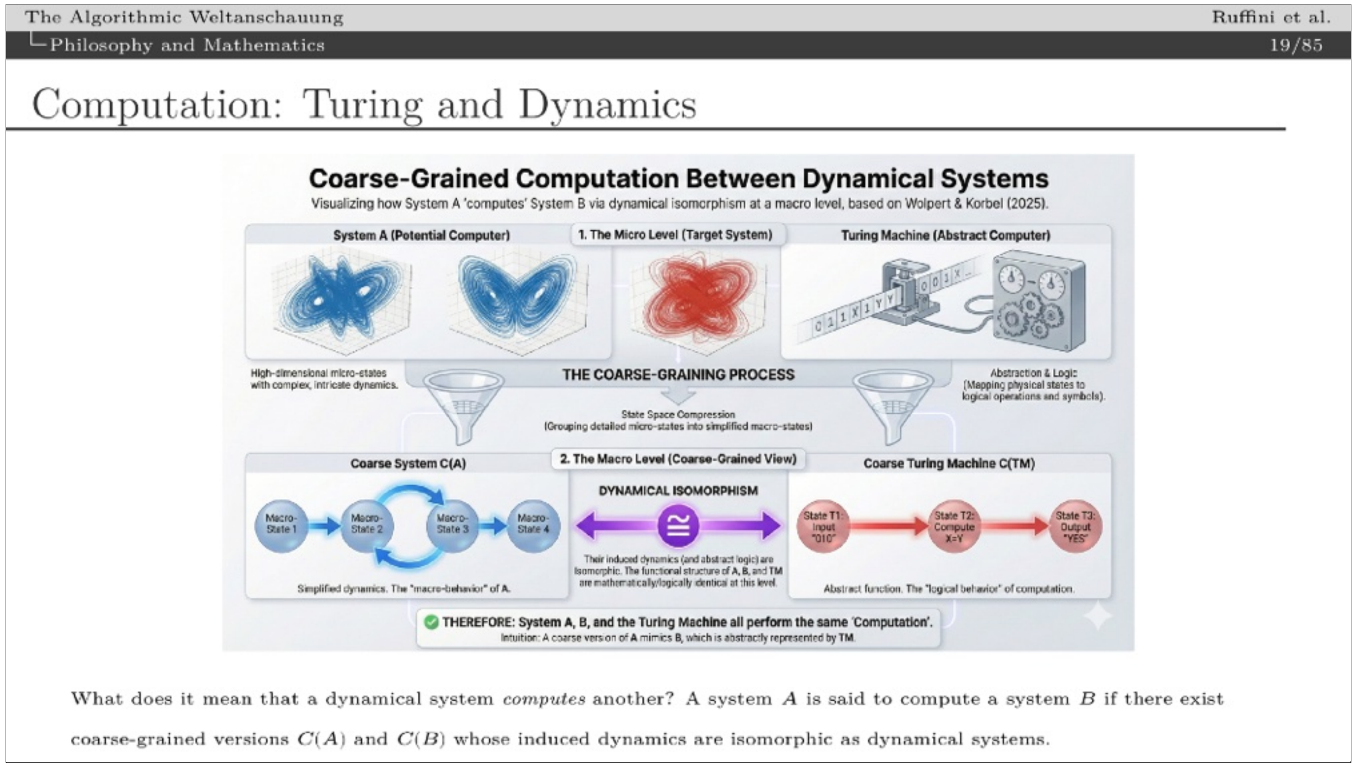
Tiling and time/computation

Here we visualize such a tiling metaphorically. On the left, it is a Penrose tiling. It is made up of a few types of pieces that fit in a specific way, and they can give rise to beautiful, complex patterns—apparent complexity emerging from inherent simplicity.

If we can slice the tiling along some curves (in white) and derive emergent rules to transition from one slice to the other, at least locally, we can recover the notion of time… and computation. On the right, we have another critical example of tiling using Wang tiles. This tiling system is a universal computer, i.e., capable of universal computation.


THE LOGIC & BITS FOUNDATION (METAMATHEMATICS)
p→q
∀x
∃y
¬q → ¬p
A→B
COARSE-GRAINING
Groups
ϕ(x)
Manifolds
THE MATHEMATICAL STRUCTURE (ABSTRACT SYSTEMS)
Sets
Groups
Tensors
F = ma
d/dx
E = mc²
EMERGENT PHYSICAL LAWS (SPECIFIC EQUATIONS)
EMERGENCE
Evolution
Genetics
Heart
Genetics
Metabolism
EMERGENT BIOLOGY (COMPLEX SYSTEMS)
Homeostasis
FROM LOGIC & BITS TO MATHEMATICS TO PHYSICAL LAWS TO BIOLOGY

There is a natural construction that starts from mathematics and information (algorithms), then moves into time, computation, and physics, and finally gives rise to emergent computational forms such as agents and life.



The Algorithmic Weltanschauung — Ruffini et al.

└Philosophy and Mathematics — 19/85

Computation: Turing and Dynamics

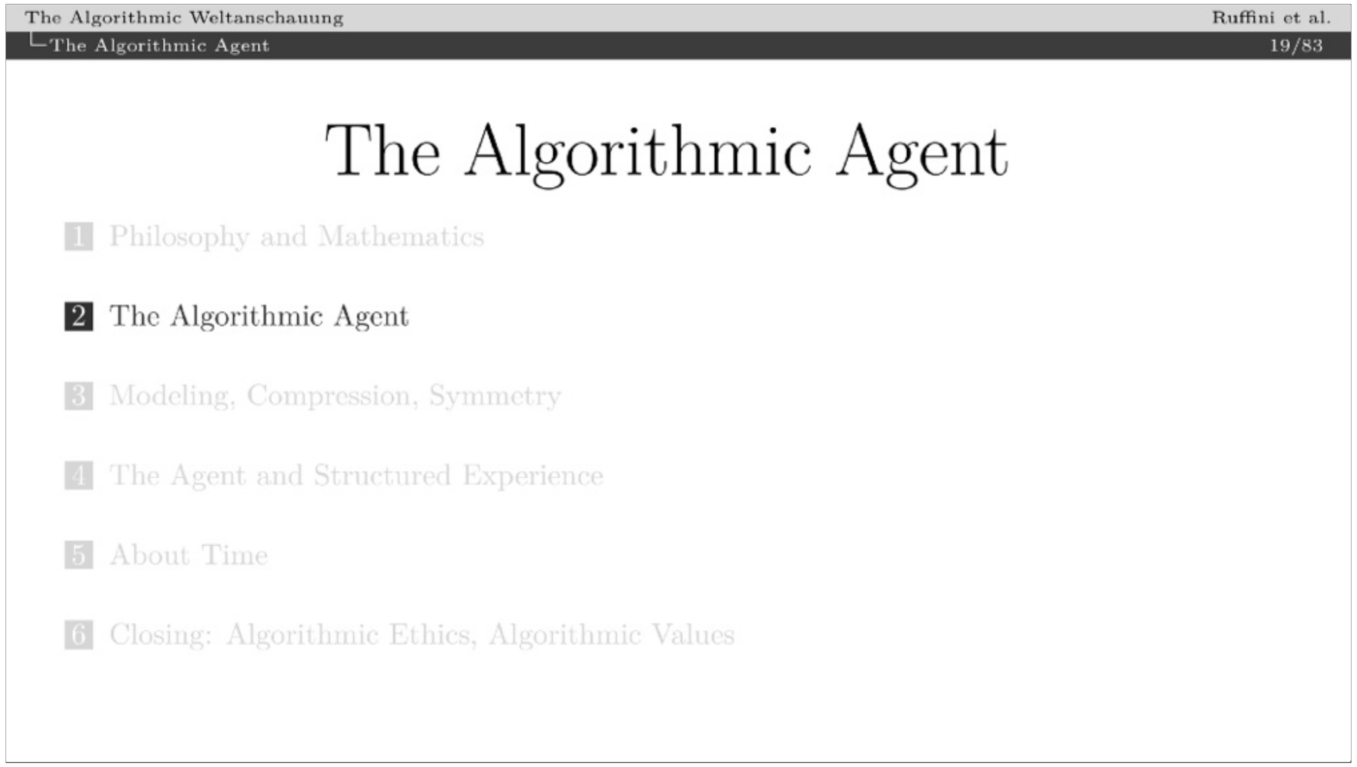
Coarse-Grained Computation Between Dynamical Systems
Visualizing how System A 'computes' System B via dynamical isomorphism at a macro level, based on Wolpert & Korbel (2025).
System A (Potential Computer)
1. The Micro Level (Target System)
Turing Machine (Abstract Computer)
0 1 1 X 1 Y Y 0 0 1 X ...
High-dimensional micro-states with complex, intricate dynamics.
THE COARSE-GRAINING PROCESS
State Space Compression (Grouping detailed micro-states into simplified macro-states)
Abstraction & Logic (Mapping physical states to logical operations and symbols).
Coarse System C(A)
2. The Macro Level (Coarse-Grained View)
Coarse Turing Machine C(TM)
Macro-State 1
Macro-State 2
Macro-State 3
Macro-State 4
DYNAMICAL ISOMORPHISM
≅
State T1: Input "010"
State T2: Compute X=Y
State T3: Output "YES"
Simplified dynamics. The "macro-behavior" of A.
Their induced dynamics (and abstract logic) are isomorphic. The functional structure of A, B, and TM are mathematically/logically identical at this level.
Abstract function. The "logical behavior" of computation.
✓ THEREFORE: System A, B, and the Turing Machine all perform the same 'Computation'.
Intuition: A coarse version of A mimics B, which is abstractly represented by TM.

What does it mean that a dynamical system *computes* another? A system A is said to compute a system B if there exist coarse-grained versions $C(A)$ and $C(B)$ whose induced dynamics are isomorphic as dynamical systems.

To ground this argument, we must rigorously define our terms. Specifically, what does it mean for a physical system to compute?

Drawing on the formalisms of Wolpert and Korbel (2025), we adopt a precise mathematical definition: a physical system performs computation if, at a coarse-grained or macroscopic level, its dynamics are isomorphic to a Turing machine. Under this definition, computation is established when physical macro-states map reliably to logical operations. This framework further implies that one dynamical system can 'compute'—or emulate—another through a similar mapping of states.

Current evidence suggests that all physical systems are computable in this sense. This observation constitutes the core of pancomputationalism and digital physics—the hypothesis that the universe is fundamentally computational in nature.



The Algorithmic Agent

1 Philosophy and Mathematics

2 The Algorithmic Agent

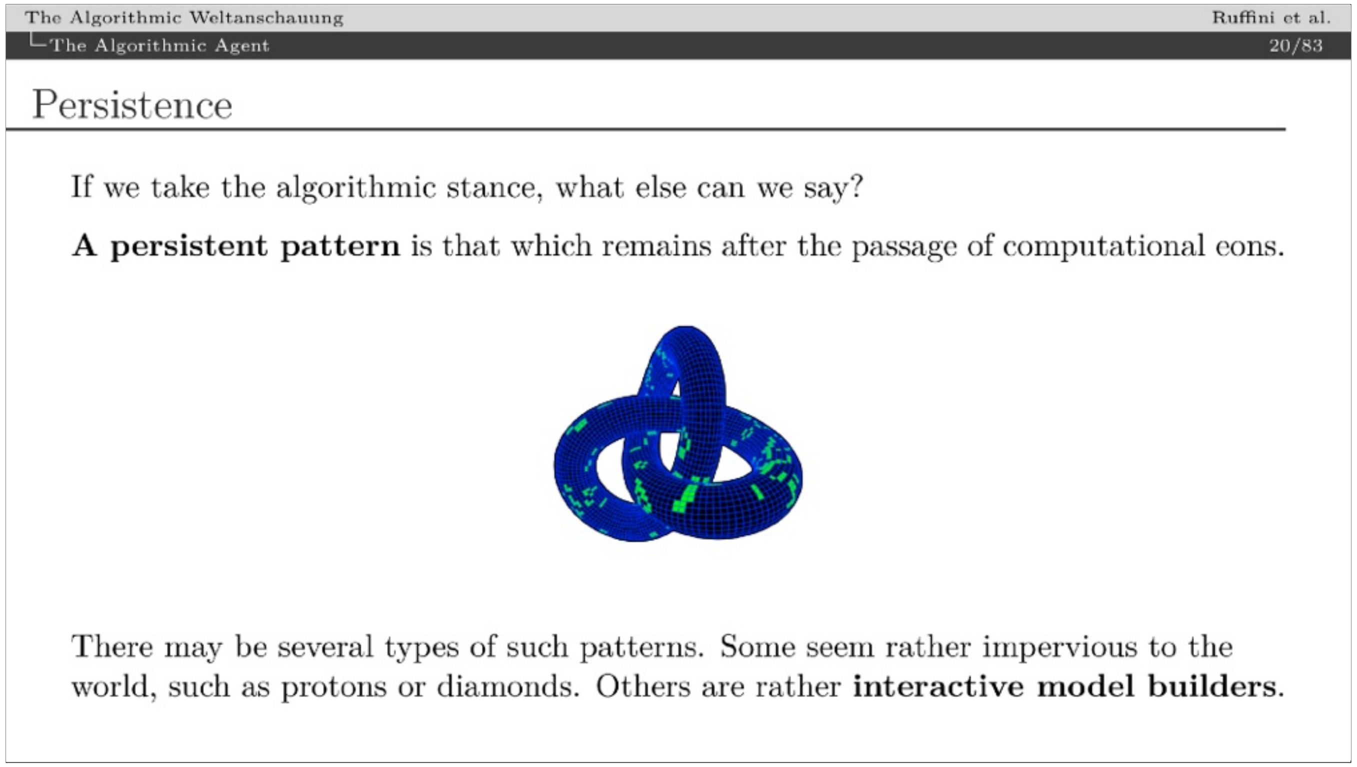
3 Modeling, Compression, Symmetry

4 The Agent and Structured Experience

5 About Time

6 Closing: Algorithmic Ethics, Algorithmic Values

Let us now move from time and computation to the Algorithmic Agent.



Persistence

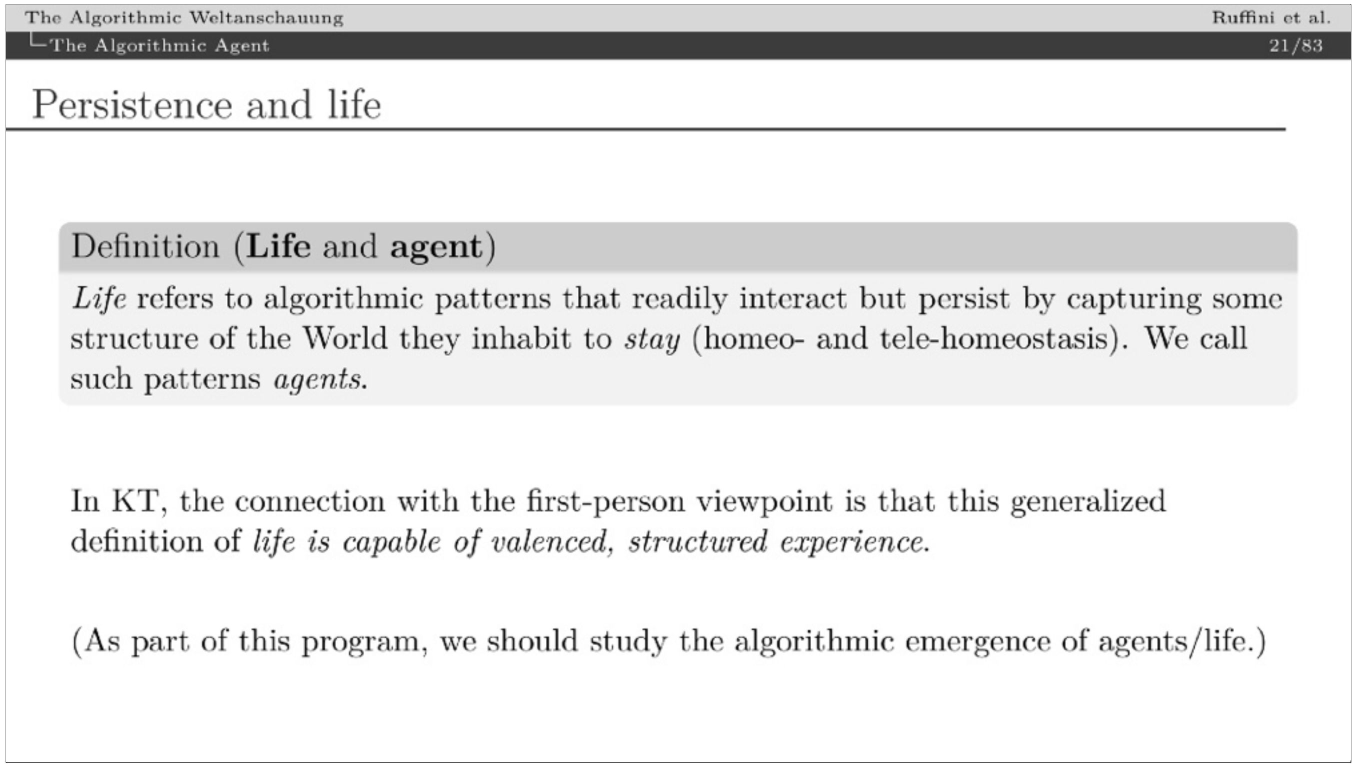
If we take the algorithmic stance, what else can we say?

A persistent pattern is that which remains after the passage of computational eons.

There may be several types of such patterns. Some seem rather impervious to the world, such as protons or diamonds. Others are rather **interactive model builders**.

The central hypothesis is that in an algorithmic or computational soup, under some conditions, there will be emerging persistent patterns (such as in the game of life in the picture, or as those found in Lenia). That is, in this soup, some patterns persist longer than others.

Persistence does not mean that an individual pattern persists. Perhaps it reproduces and evolves slowly. This is also a form of algorithmic persistence. Thus, while some patterns may be static, such as protons or diamonds, others are interactive model builders and replicators.



Persistence and life

Definition (Life and agent)

Life refers to algorithmic patterns that readily interact but persist by capturing some structure of the World they inhabit to *stay* (homeo- and tele-homeostasis). We call such patterns *agents*.

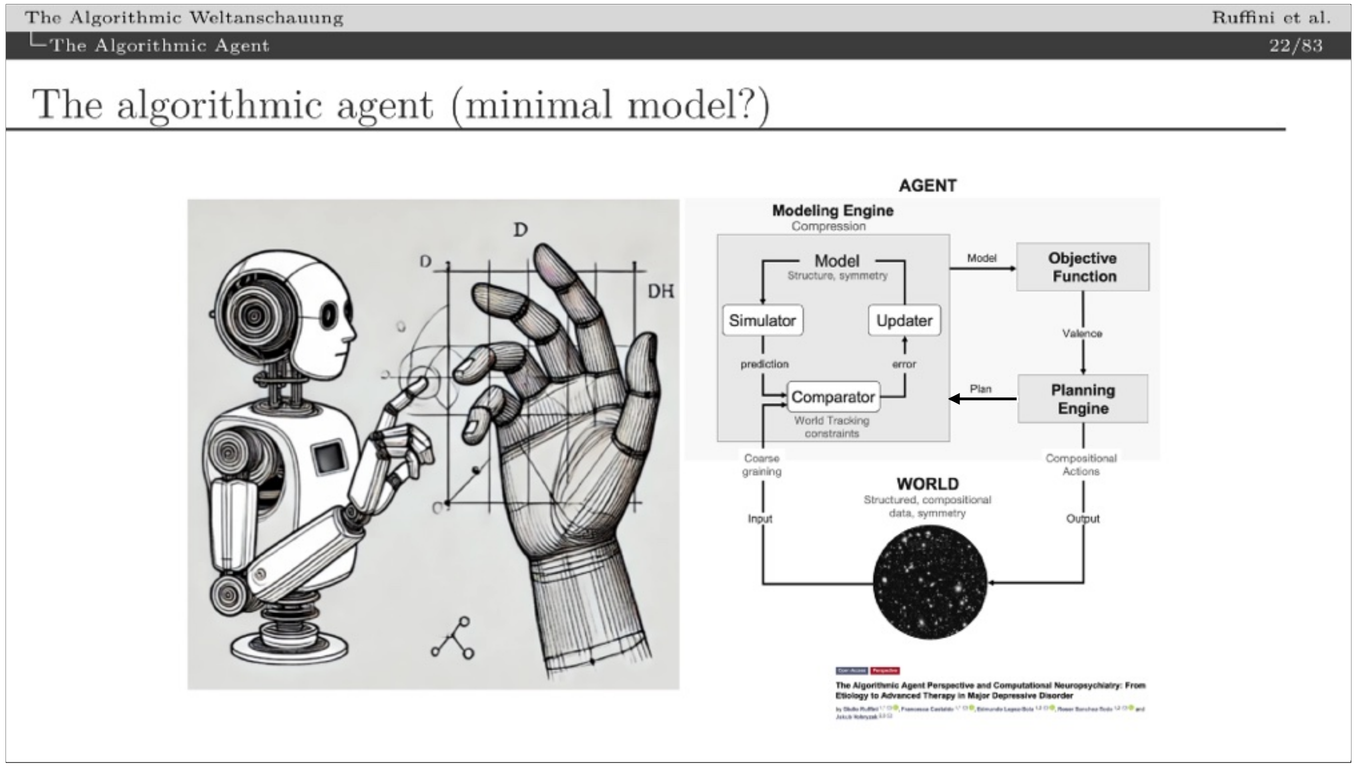
In KT, the connection with the first-person viewpoint is that this generalized definition of *life is capable of valenced, structured experience.*

(As part of this program, we should study the algorithmic emergence of agents/life.)

We thus define ‘Living form' or 'Agent' as an algorithmic pattern that persists by capturing the structure of the world to actively maintain homeostasis — by planning and acting on the world.

In our framework, this “algorithmic structure” of the agent is where valenced, structured experience takes place.

The agent is the algorithmic basis of the structured experience we know as living beings.



The algorithmic agent (minimal model?)

Open Access Perspective

The Algorithmic Agent Perspective and Computational Neuropsychiatry: From Etiology to Advanced Therapy in Major Depressive Disorder

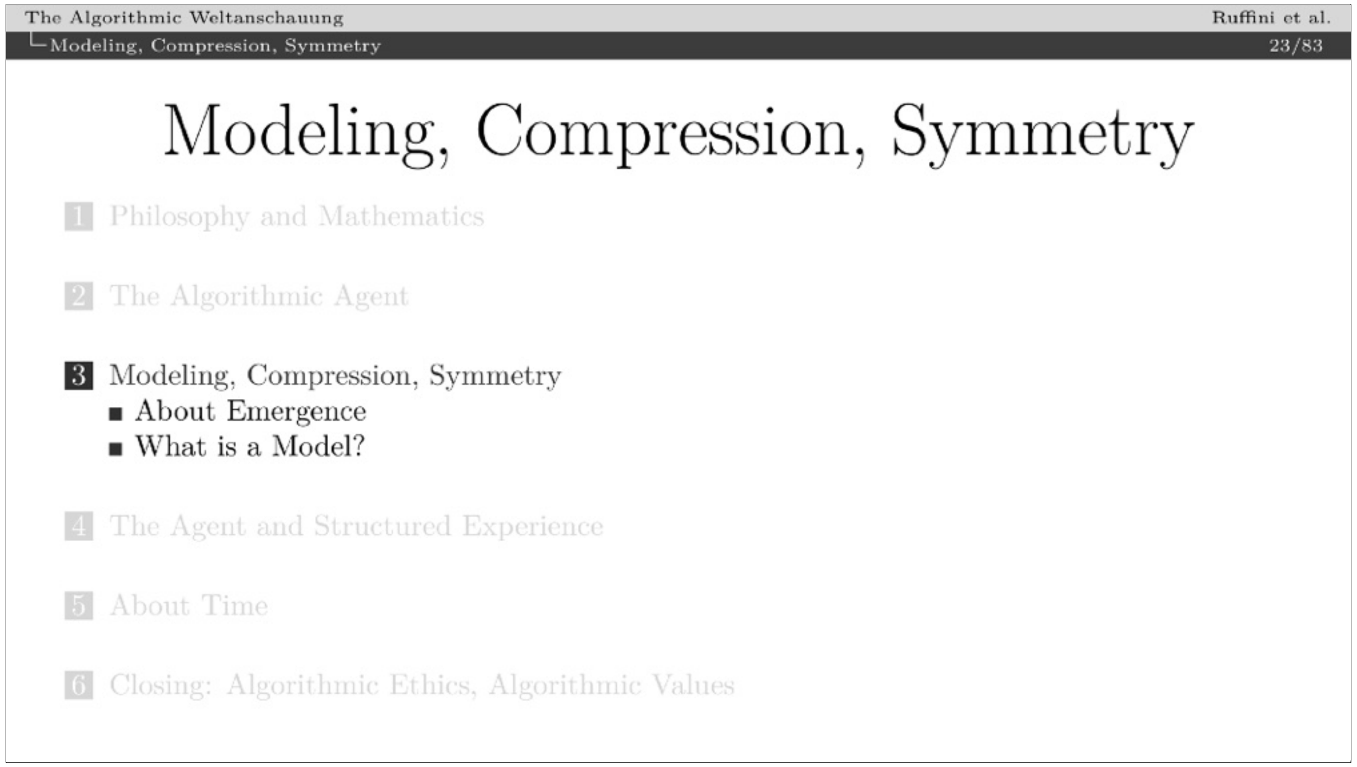
by Giulio Ruffini, Francesca Castaldo, Edmundo Lopez-Sola, Roser Sanchez-Todo and Jakub Vohryzek

What is an algorithmic agent? A program.

There is a minimal model of such an agent (and of Life). It consists of a Modeling Engine (to compress data), a Planning Engine (to simulate futures), and an Objective Function (providing valence, or ‘value'), the ultimate driver of actions by the agent. The agent interacts in a loop with the world: predicting future world and valence (internal) states, acting, and updating its world/self model.

I propose that this is a *minimal* version of an agent; all the modules are necessary. In part, this is justified by the Regulator theorem, which I will discuss later.

Furthermore, the definition is agnostic on the physical implementation of this computational system: carbon-based, silicon, quark-based… This feature is part of the framework's potential, which applies readily to all life forms, exobiology, and AI.



Modeling, Compression, Symmetry

1 Philosophy and Mathematics

2 The Algorithmic Agent

3 Modeling, Compression, Symmetry
- About Emergence
- What is a Model?

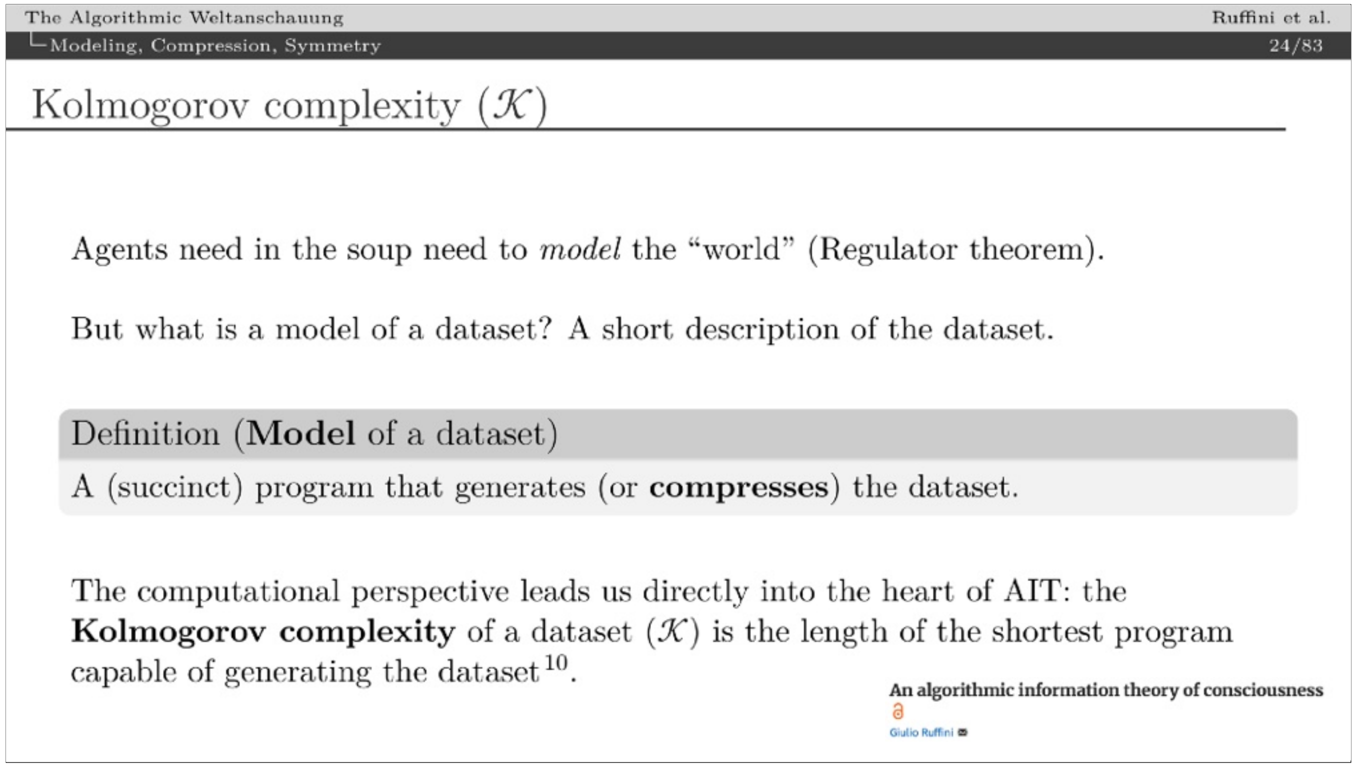
4 The Agent and Structured Experience

5 About Time

6 Closing: Algorithmic Ethics, Algorithmic Values

We now shift to discussing the Modeling Engine in a bit more detail.

What is a model in the algorithmic/mathematical context? We will see it reduces to the ideas of compression and symmetry – two sides of the same coin.



Kolmogorov complexity ($\mathcal{K}$)

Agents need in the soup need to *model* the "world" (Regulator theorem).

But what is a model of a dataset? A short description of the dataset.

Definition (Model of a dataset)

A (succinct) program that generates (or **compresses**) the dataset.

The computational perspective leads us directly into the heart of AIT: the **Kolmogorov complexity** of a dataset ($\mathcal{K}$) is the length of the shortest program capable of generating the dataset[10].

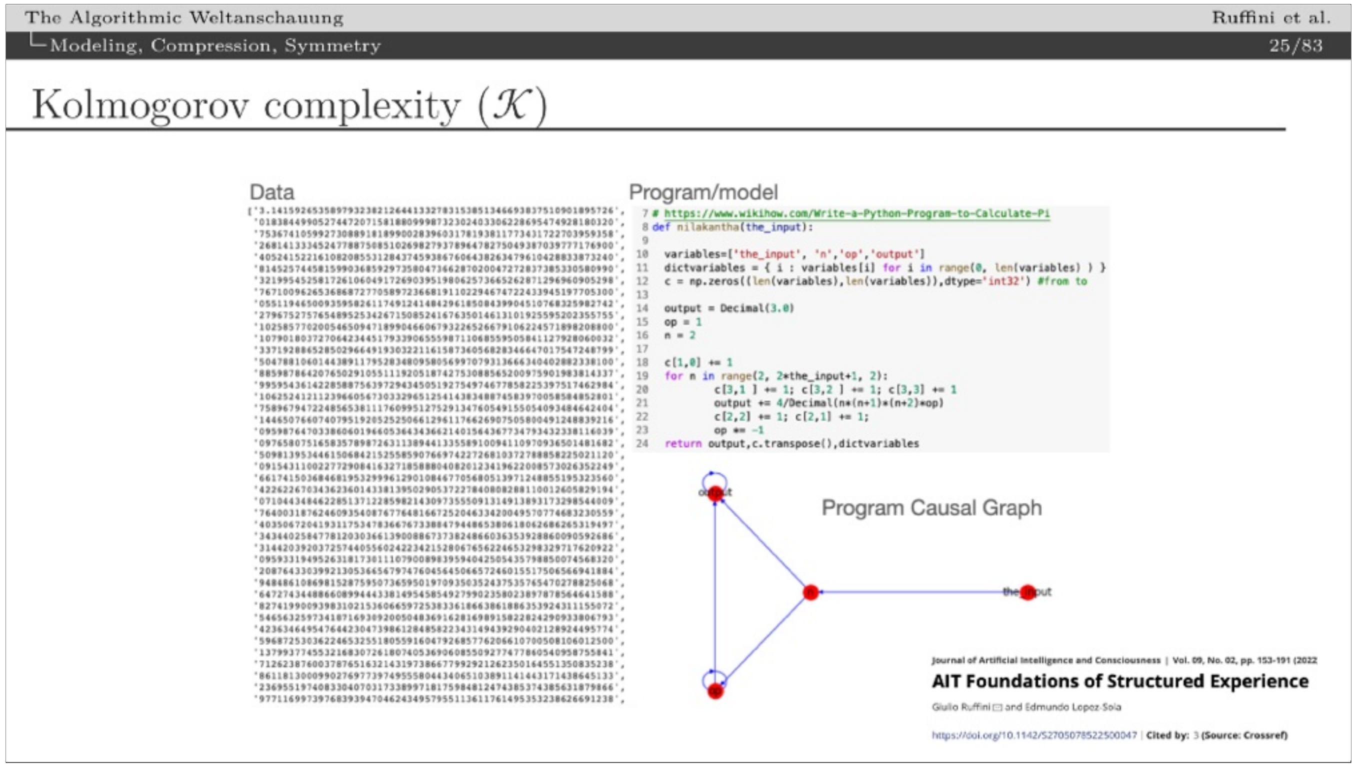
An algorithmic information theory of consciousness

Giulio Ruffini

In the algorithmic framework, where everything is "program, the central conceptual pillar is that of algorithmic or Kolmogorov complexity. It builds on the work of Turing and the Turing machine, or other similar frameworks of computation.

Recall that computation is the idea of executing a series of procedures in steps.

We talk of data objects, and we inquire what programs can produce them. The Kolmogorov complexity or algorithmic information of an object is the length of the shortest program capable of generating the dataset. Programs are thus data compressors. Agents need such programs to compress world data, which we also call "models" of the dataset. This allows them to understand the world, reason, and predict the future, and act to maximize their goal (telehomeostasis).



Kolmogorov complexity ($\mathcal{K}$)

Data

```
['3.1415926535897932382126441332783153851346693837510901895726',
 '018384499052744720715818809998732302403306228695474928180320',
 '753674105992730889181899002839603178193811773431722703959358',
 '268141333452477887508510269827937896478275049387039777176900',
 '405241522161082085531284374593867606438263479610428833873240',
 '814525744581599036859297358047366287020047272837385330580990',
 '321995452581726106049172690395198062573665262871296960905298',
 '767100962653686872770589723668191102294674722433945197705300',
 '055119465009359582611749124148429618508439904510768325982742',
 '279675275765489525342671508524167635014613101925595202355755',
 '102585770200546509471899046606793226526679106224571898208800',
 '107901803727064234451793390655598711068559505841127928060032',
 '337192886528502966491930322116158736056828346647017547248799',
 '504788106014438911795283480958056997079313666340402882338100',
 '885987864207650291055111920518742753088565200975901983814337',
 '995954361422858875639729434505192754974677858225397517462984',
 '106252412112396605673033296512541438348874583970058584852801',
 '758967947224856538111760995127529134760549155054093484642404',
 '144650766074079519205252506612961176626907505800491248839216',
 '095987647033860601966053643436621401564367734793432338116039',
 '097658075165835789872631138944133558910094110970936501481682',
 '509813953446150684215255859076697422726810372788858225021120',
 '091543110022772908416327185888040820123419622008573026352249',
 '661741503684681953299961290108467705680513971248855195323560',
 '422622670343623601433813950290537227840808288110012605829194',
 '071044348462285137122859821430973555091314913893173298544009',
 '764003187624609354087677648166725204633420049570774683230559',
 '403506720419311753478366767338847944865380618062686265319497',
 '343440258477812030366139008867373824866036353928860090592686',
 '314420392037257440556024223421528067656224653298329717620922',
 '095933194952631817301110790089839594042505435798850074568320',
 '208764330399213053665679747604564506657246015517506566941884',
 '948486108698152875950736595019709350352437535765470278825068',
 '647274344886608994443381495458549279902358023897878564641588',
 '827419900939831021536066597253833618663861886353924311155072',
 '546563259734187169309200504836916281698915822824290933806793',
 '423634649547644230473986128485822343149439290402128924495774',
 '596872530362246532551805591604792685776206610700508106012500',
 '137993774553216830726180740536906085509277477860540958755841',
 '712623876003787651632143197386677992921262350164551350835238',
 '861181300099027697739749555804434065103891141443171438645133',
 '236955197408330407031733899718175984812474385374385631879866',
 '977116997397683939470462434957955113611761495353238626691238',
```

Program/model

```python
# https://www.wikihow.com/Write-a-Python-Program-to-Calculate-Pi
def nilakantha(the_input):

  variables=['the_input', 'n','op','output']
  dictvariables = { i : variables[i] for i in range(0, len(variables) ) }
  c = np.zeros((len(variables),len(variables)),dtype='int32') #from to

  output = Decimal(3.0)
  op = 1
  n = 2

  c[1,0] += 1
  for n in range(2, 2*the_input+1, 2):
          c[3,1 ] += 1; c[3,2 ] += 1; c[3,3] += 1
          output += 4/Decimal(n*(n+1)*(n+2)*op)
          c[2,2] += 1; c[2,1] += 1;
          op *= -1
  return output,c.transpose(),dictvariables
```

Program Causal Graph

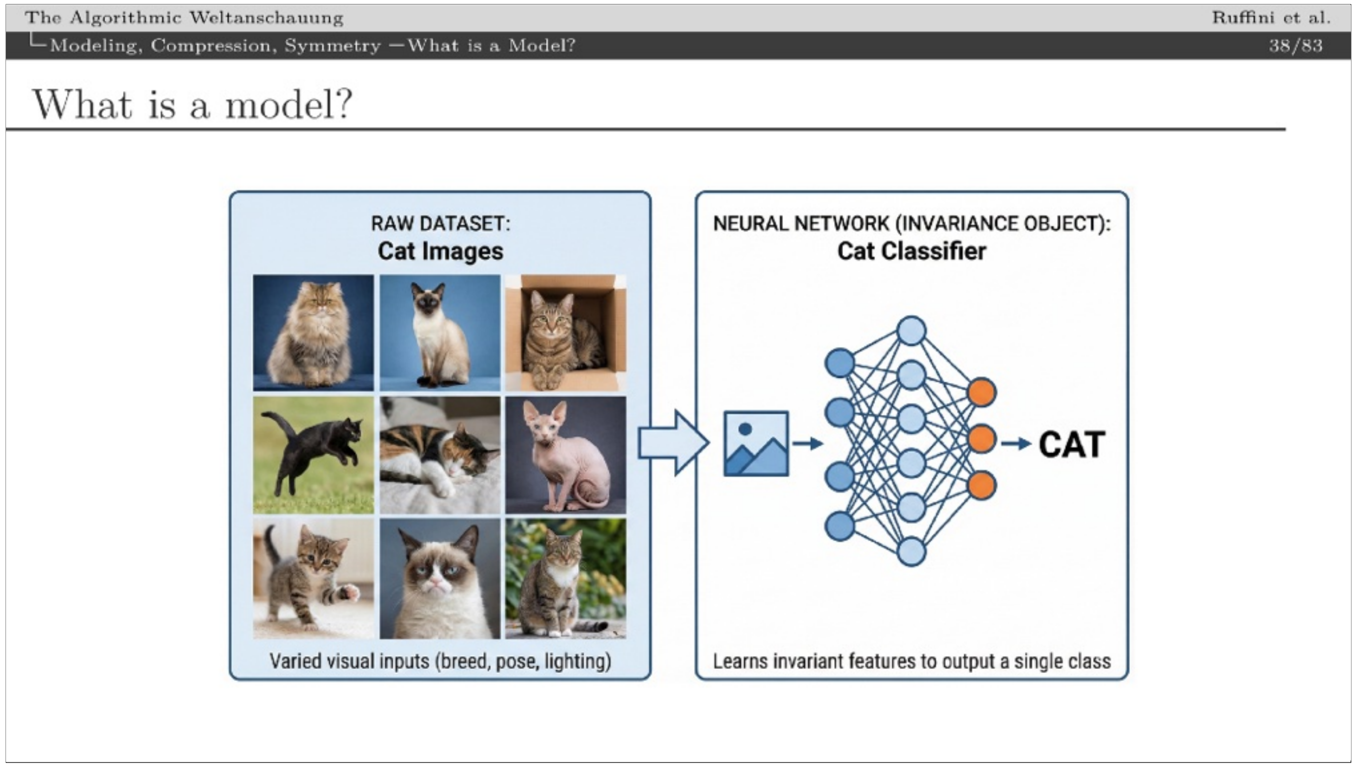
Journal of Artificial Intelligence and Consciousness | Vol. 09, No. 02, pp. 153-191 (2022

AIT Foundations of Structured Experience

Giulio Ruffini and Edmundo Lopez-Sola

https://doi.org/10.1142/S2705078522500047 | **Cited by:** 3 **(Source: Crossref)**

Here's an extreme example. On the left you have a series of digits, which you may recognize as the number pi. They are random-looking, and in classical Shannon entropy terms, their entropy or information content is maximal. Yet, in fact, they are generated by a very short program (on the right, in Python code). The algorithmic information content of the dataset (which we may make as large as we wish) is very small! Once we have a short program that generates the data, we may inquire about its structure. In the bottom right, you have an attempt at capturing the structure, at least the relationship between the variables that the program uses.



What is a model?

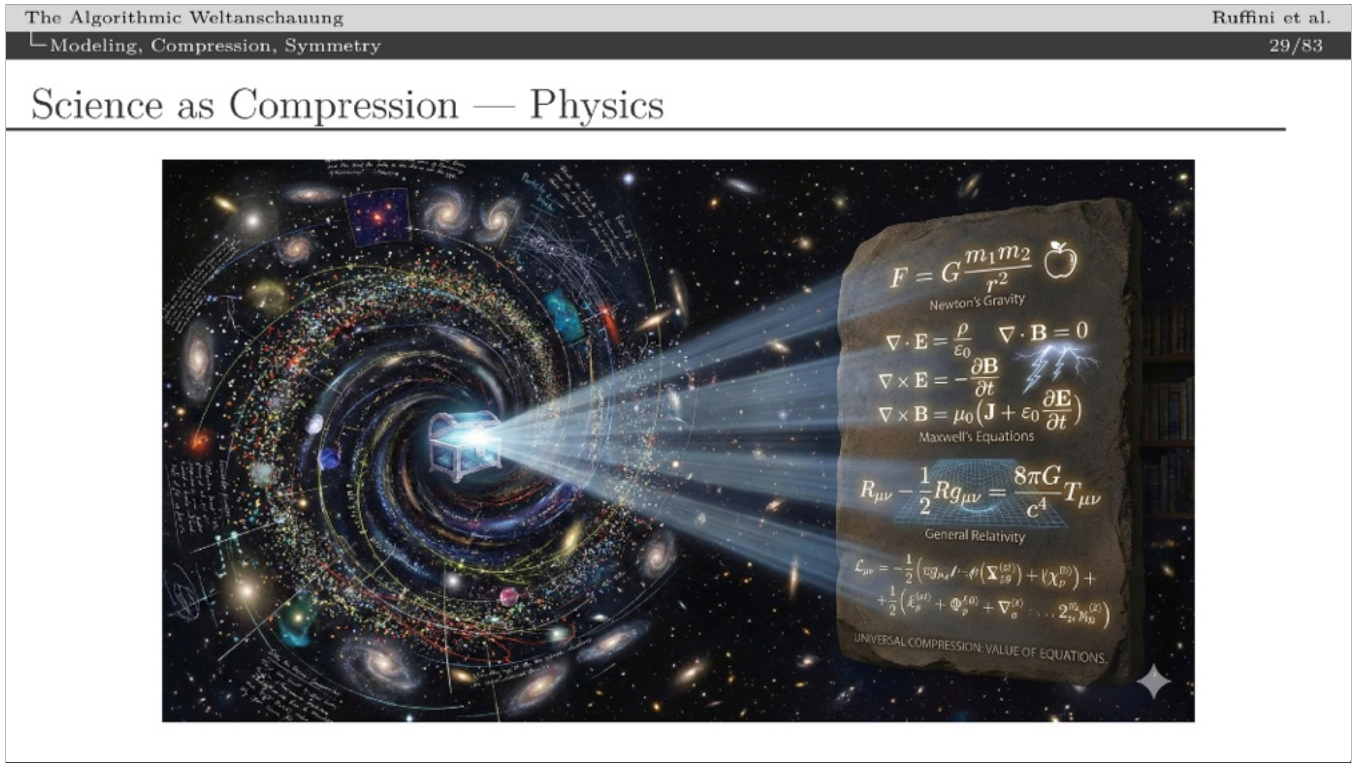
RAW DATASET:
Cat Images
Varied visual inputs (breed, pose, lighting)
NEURAL NETWORK (INVARIANCE OBJECT):
Cat Classifier
CAT
Learns invariant features to output a single class

Here’s another example of a program embedding a world model, i.e., one that can help compress data. It is a program that classifies images and is capable of telling you which ones contain the image of a cat.

If you feed it a dataset consisting of a stack of 1E9 images of cats and a few dogs, it will tell you that most of them are images of cats.

You can then use that knowledge to describe the dataset to someone else more efficiently than the raw pixels of the billions of images. It may not be obvious, but knowing that an image contains a cat allows you to write a short program with a few parameters to describe the image, at least much shorter than if you don’t have that information.

The space of all possible 1000x1000 pixel images at 32 bits per pixel is huge! The subspace of images with cats is much, much smaller.



The Algorithmic Weltanschauung Ruffini et al.

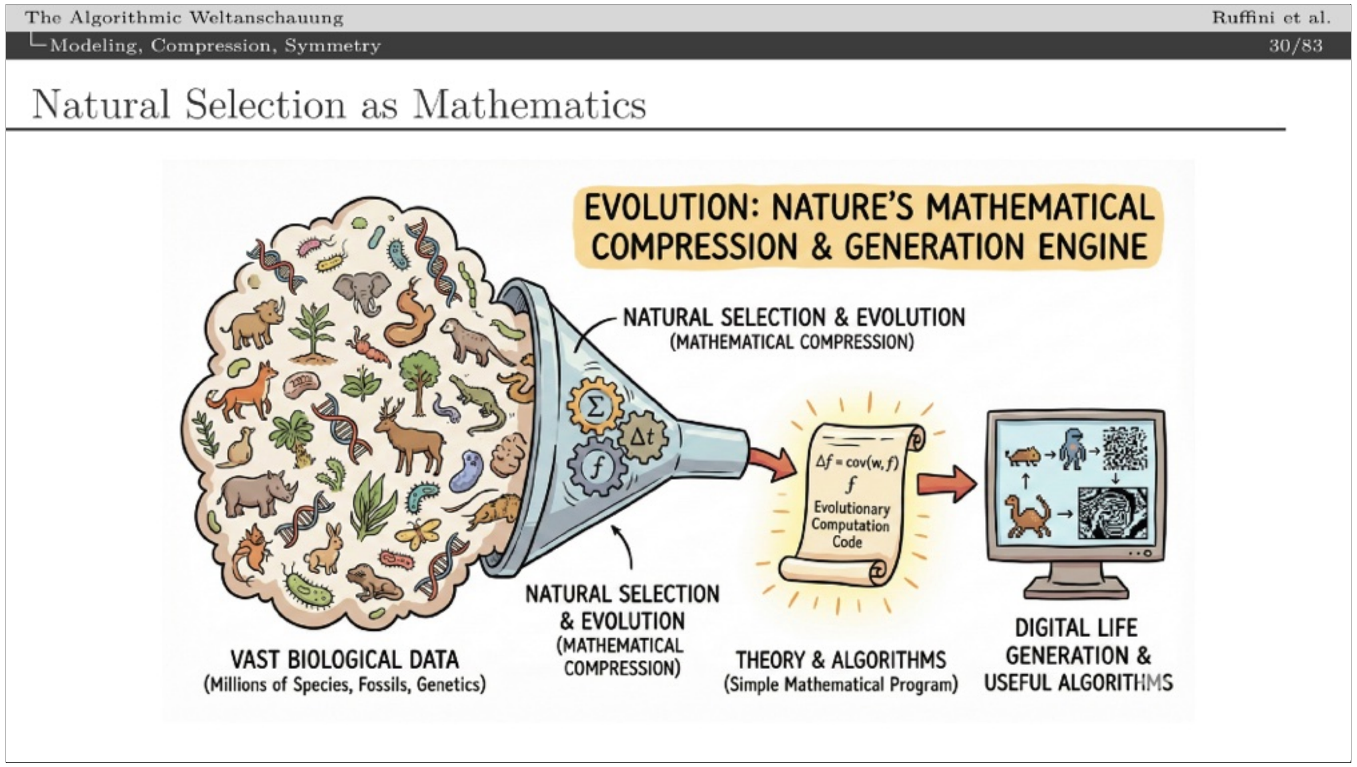
└Modeling, Compression, Symmetry 29/83

Science as Compression — Physics

$F = G\frac{m_1 m_2}{r^2}$
Newton's Gravity
$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$ $\nabla \cdot \mathbf{B} = 0$
$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$
$\nabla \times \mathbf{B} = \mu_0\left(\mathbf{J} + \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}\right)$
Maxwell's Equations
$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}$
General Relativity
UNIVERSAL COMPRESSION: VALUE OF EQUATIONS.

In fact, writing short programs to describe large amounts of data is what *science* is all about. That is what we mean by finding patterns and structure in the data. And this is why mathematics is central to science, as the “science of structure”.

Among the major feats of humankind is the discovery of such programs, which we call laws, such as the laws of gravity or electromagnetism or the standard model. There are, in fact, programs that summarize and compress vast amounts of data into a few lines of mathematics. They can then be used to “calculate” or “simulate” physical phenomena, and they provide the foundation on which our technology is built.

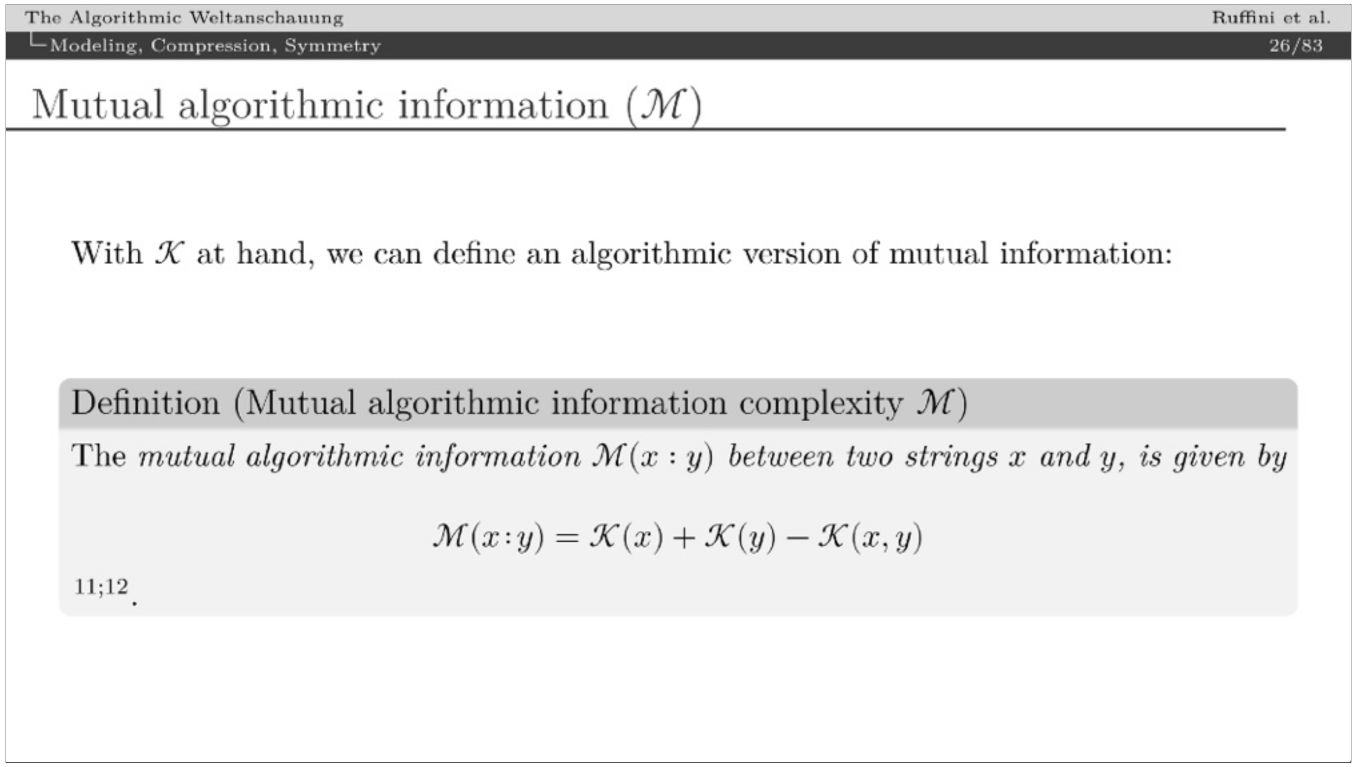


Natural Selection as Mathematics

EVOLUTION: NATURE'S MATHEMATICAL COMPRESSION & GENERATION ENGINE
NATURAL SELECTION & EVOLUTION
(MATHEMATICAL COMPRESSION)
Σ
Δt
f
Δf = cov(w, f)
f
Evolutionary Computation Code
VAST BIOLOGICAL DATA
(Millions of Species, Fossils, Genetics)
NATURAL SELECTION & EVOLUTION
(MATHEMATICAL COMPRESSION)
THEORY & ALGORITHMS
(Simple Mathematical Program)
DIGITAL LIFE GENERATION & USEFUL ALGORITHMS

This view of science as compression applies not only to physics or engineering; it applies to biology and all the budding sciences. As we discover laws and patterns in world data, we can compress and simulate vast amounts of data.

The law of natural selection and evolution in biology is a powerful example of the discovery of structure in the life sciences. These laws or models are also programs; they are 100% mathematical in nature. They capture structure and can be used to compress natural data (“understand”) and simulate scenarios.



Mutual algorithmic information ($\mathcal{M}$)

With $\mathcal{K}$ at hand, we can define an algorithmic version of mutual information:

Definition (Mutual algorithmic information complexity $\mathcal{M}$)

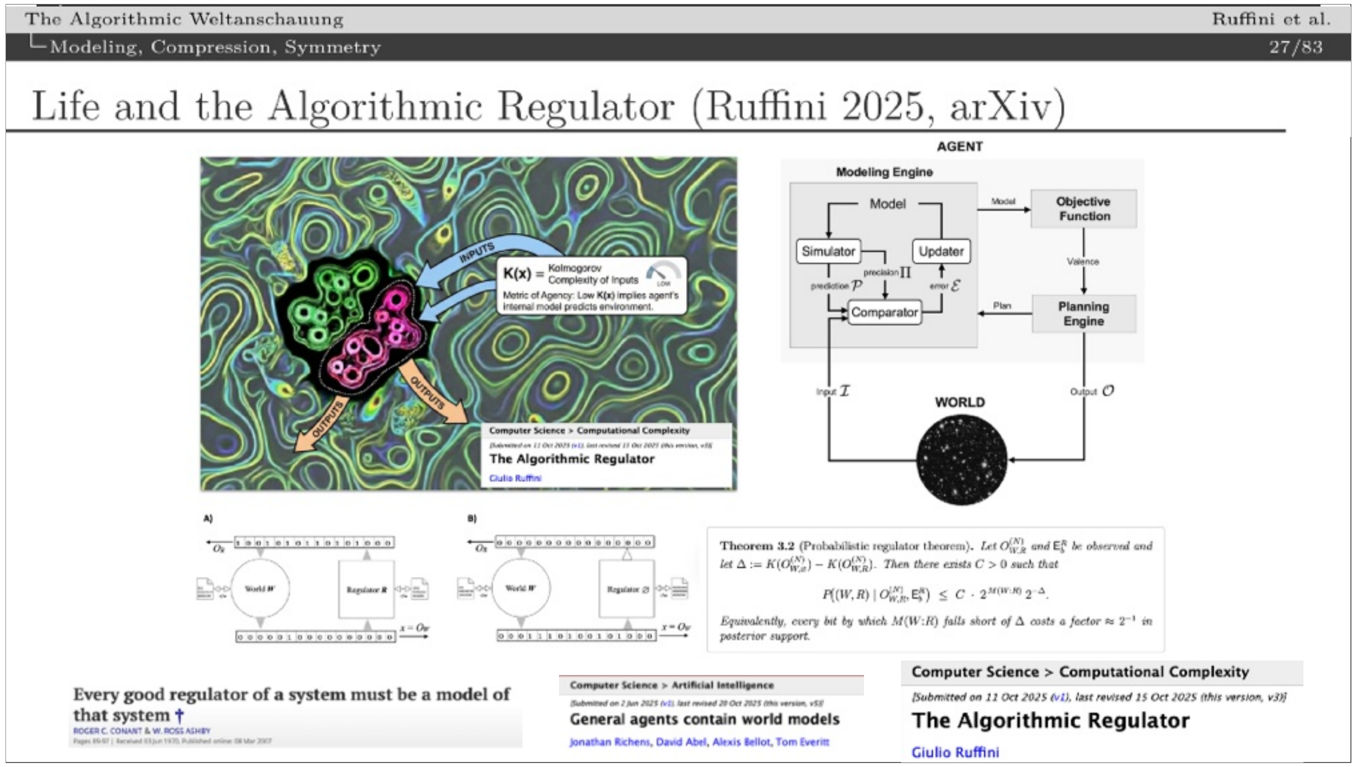
The *mutual algorithmic information* $\mathcal{M}(x:y)$ *between two strings* x *and* y*, is given by*

$$\mathcal{M}(x:y) = \mathcal{K}(x) + \mathcal{K}(y) - \mathcal{K}(x,y)$$

[11;12].

A key element in our discussion henceforth is a natural derivative of Kolmogorov complexity: mutual algorithmic information.

This definition captures the idea that similar programs lurk beneath different data strings. It describes the shared algorithmic structure of the data generators. Thus, if two strings have high mutual algorithmic information, you can find common algorithmic structures in the programs that generate them.



The Algorithmic Weltanschauung — Ruffini et al.

Modeling, Compression, Symmetry — 27/83

Life and the Algorithmic Regulator (Ruffini 2025, arXiv)

Theorem 3.2 (Probabilistic regulator theorem). *Let $O^{(N)}_{W,R}$ and E^R_S be observed and let $\Delta := K(O^{(N)}_{W,\varnothing}) - K(O^{(N)}_{W,R})$. Then there exists $C > 0$ such that*

$$P((W,R) \mid O^{(N)}_{W,R}, \mathsf{E}^R_S) \;\le\; C \cdot 2^{M(W:R)}\, 2^{-\Delta}.$$

Equivalently, every bit by which $M(W{:}R)$ falls short of Δ costs a factor $\approx 2^{-1}$ in posterior support.

Every good regulator of a system must be a model of that system †

ROGER C. CONANT & W. ROSS ASHBY

Pages 89-97 | Received 03 Jun 1970, Published online: 08 Mar 2007

Computer Science > Artificial Intelligence

[Submitted on 2 Jun 2025 (v1), last revised 20 Oct 2025 (this version, v5)]

General agents contain world models

Jonathan Richens, David Abel, Alexis Bellot, Tom Everitt

Computer Science > Computational Complexity

[Submitted on 11 Oct 2025 (v1), last revised 15 Oct 2025 (this version, v3)]

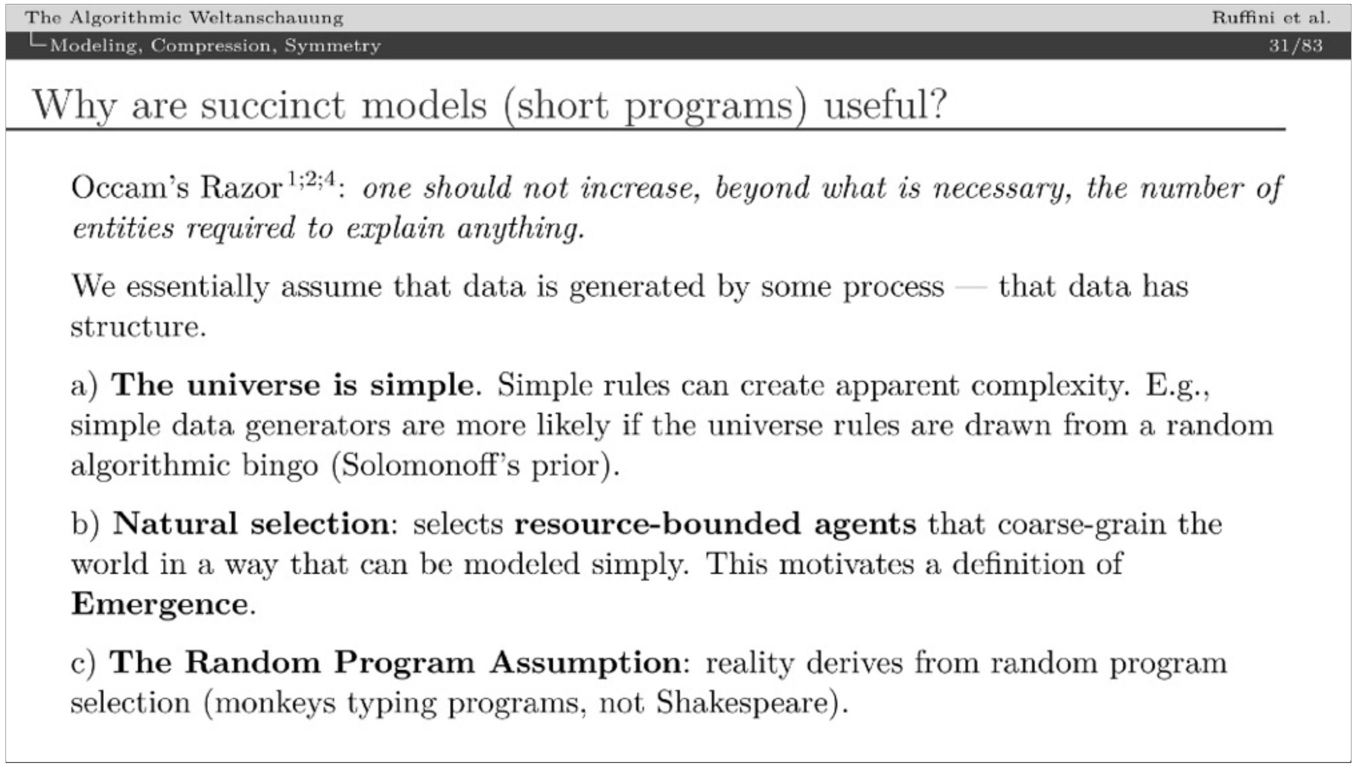
The Algorithmic Regulator

Giulio Ruffini

This concept is central to thinking about life and how to detect algorithmic agents out there.

Recall that agents need to discover models of the world to go about their business of surviving as individuals or patterns (i.e., reproducing). We can quantify this using the concept of mutual algorithmic information: agents must share algorithmic information with the world! That's precisely what it means to have a useful model of a dataset that another model generates. The two models (in the agent and the one generating world data) need to have common elements.

In a recent paper, I argue that this may be a useful way to discover life. First, it proposes that a successful agent must be capable of "simplifying" the inputs it receives from the world (think of a thermostat). Second, it shows that this implies that it has discovered a piece of the world model. This is the algorithmic version of the Good Regulator Theorem, which says loosely that a system capable of regulating something must have a model of the world influencing it. It is formalized by "a good regulator has to have non-trivial mutual algorithmic information with the world".



Why are succinct models (short programs) useful?

Occam's Razor[1;2;4]: *one should not increase, beyond what is necessary, the number of entities required to explain anything.*

We essentially assume that data is generated by some process — that data has structure.

a) **The universe is simple**. Simple rules can create apparent complexity. E.g., simple data generators are more likely if the universe rules are drawn from a random algorithmic bingo (Solomonoff's prior).

b) **Natural selection**: selects **resource-bounded agents** that coarse-grain the world in a way that can be modeled simply. This motivates a definition of **Emergence**.

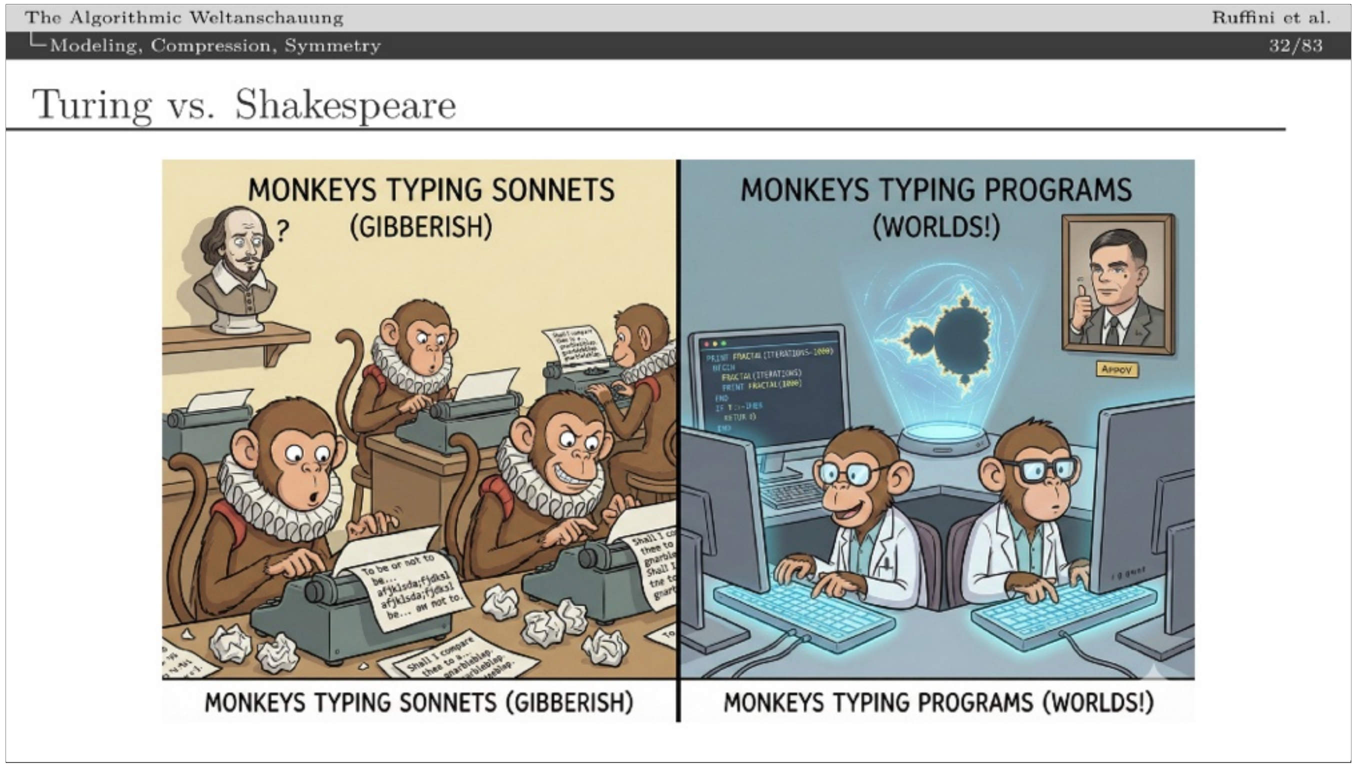
c) **The Random Program Assumption**: reality derives from random program selection (monkeys typing programs, not Shakespeare).

Ok, but why are *short* programs good world models? This intuition has a long history, including Occam's razor, Leibniz, Newton, Mach, and Einstein. But why should we have a bias for simplicity in looking for explanations?

There are different angles to this, first and foremost, the idea that *the universe is apparently complex but intrinsically simple.*

Or, that, as resource-bounded agents, our only hope is that there is simplicity out there. We may find ways to compress the world using coarse-graining.

Mathematically, the bias for simplicity in world modeling can be justified if we assume that data is generated by random programs.



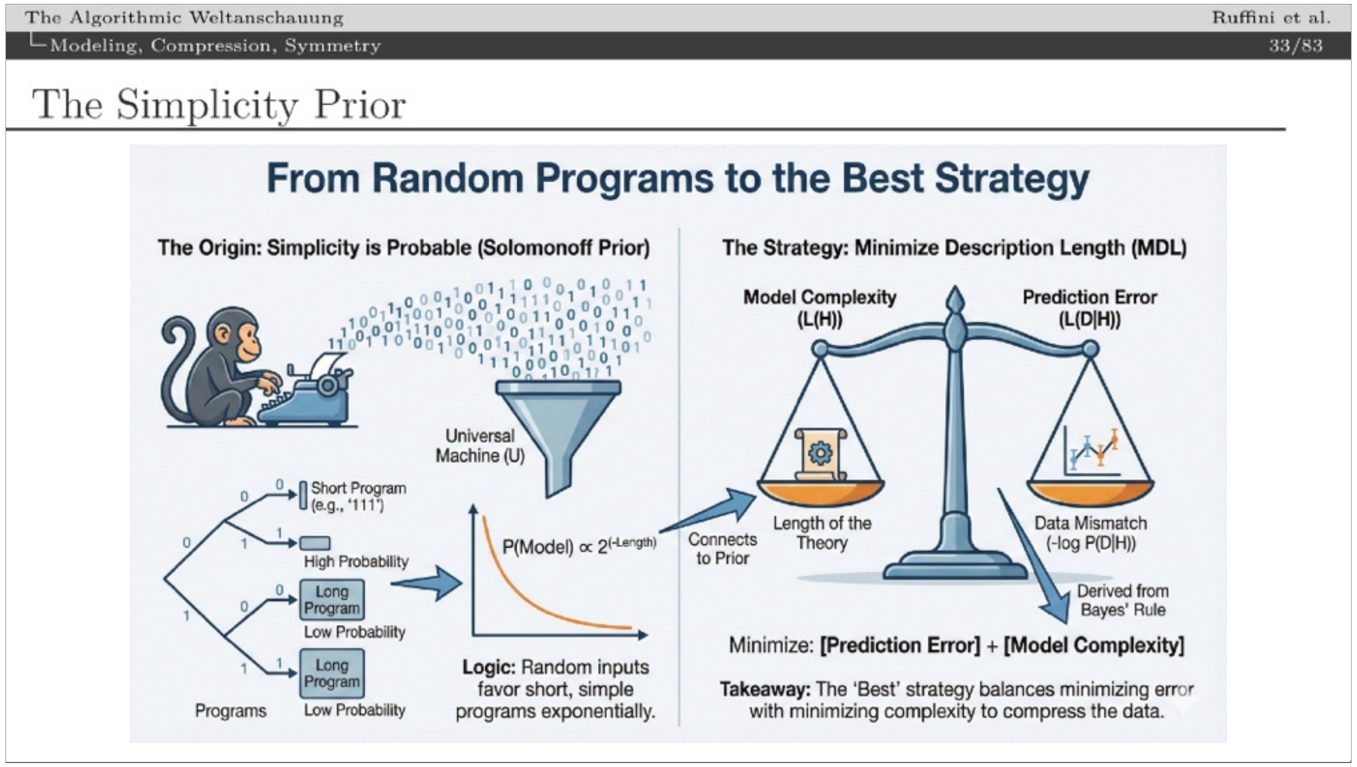
The Algorithmic Weltanschauung — Ruffini et al.

Modeling, Compression, Symmetry — 32/83

Turing vs. Shakespeare

MONKEYS TYPING SONNETS
(GIBBERISH)
?
To be or not to be... afjklsdajfjdksl afjklsdajfjdksl be... aw not to
Shall I compare thee to a...
MONKEYS TYPING PROGRAMS
(WORLDS!)
APPROV
MONKEYS TYPING SONNETS (GIBBERISH)
MONKEYS TYPING PROGRAMS (WORLDS!)

Here’s a cartoon version of this idea: if we have monkeys typing randomly, there is little probability that they will type anything like a good novel or sonnet. But if they randomly type programs, there is a much greater chance that the programs will generate beautiful structures, including the sonnets themselves! This is because the programs are structured, and structure can be described much more succinctly than truly random data. Sonnets or novels contain a lot of structure; they are not random sequences of strings.



The Simplicity Prior

From Random Programs to the Best Strategy

The Origin: Simplicity is Probable (Solomonoff Prior)

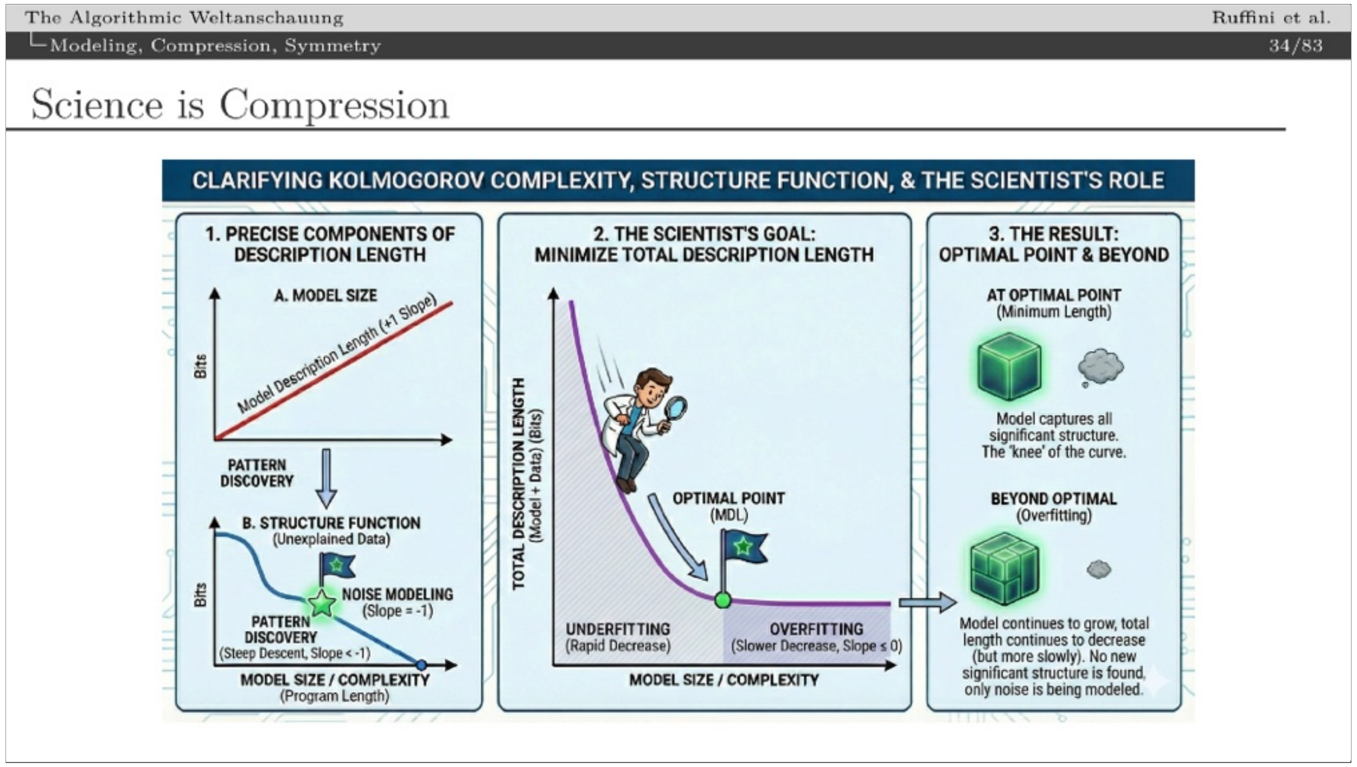
The Strategy: Minimize Description Length (MDL)

Logic: Random inputs favor short, simple programs exponentially.

Minimize: **[Prediction Error]** + **[Model Complexity]**

Takeaway: The 'Best' strategy balances minimizing error with minimizing complexity to compress the data.

And if data is generated by random programs, then one can show that searching for programs with a bias for simplicity (short programs) is the best strategy. In fact, the Minimum Description Length principle follows directly from the Solomonoff prior (the world is generated from random programs) and Bayes' rule.



Science is Compression

CLARIFYING KOLMOGOROV COMPLEXITY, STRUCTURE FUNCTION, & THE SCIENTIST'S ROLE
1. PRECISE COMPONENTS OF DESCRIPTION LENGTH
A. MODEL SIZE
Bits
Model Description Length (+1 Slope)
PATTERN DISCOVERY
B. STRUCTURE FUNCTION
(Unexplained Data)
Bits
NOISE MODELING
(Slope = -1)
PATTERN DISCOVERY
(Steep Descent, Slope < -1)
MODEL SIZE / COMPLEXITY
(Program Length)
2. THE SCIENTIST'S GOAL: MINIMIZE TOTAL DESCRIPTION LENGTH
TOTAL DESCRIPTION LENGTH
(Model + Data) (Bits)
OPTIMAL POINT
(MDL)
UNDERFITTING
(Rapid Decrease)
OVERFITTING
(Slower Decrease, Slope ≤ 0)
MODEL SIZE / COMPLEXITY
3. THE RESULT: OPTIMAL POINT & BEYOND
AT OPTIMAL POINT
(Minimum Length)
Model captures all significant structure. The 'knee' of the curve.
BEYOND OPTIMAL
(Overfitting)
Model continues to grow, total length continues to decrease (but more slowly). No new significant structure is found, only noise is being modeled.

This perspective brings us back to the scientific method, viewed here as a search for algorithmic simplicity. The scientist's objective is to identify programs—serving as models or theories—that simultaneously maximize fidelity to observed data and minimize complexity. This optimization problem is rigorously formalized by the principle of Minimum Description Length (MDL).

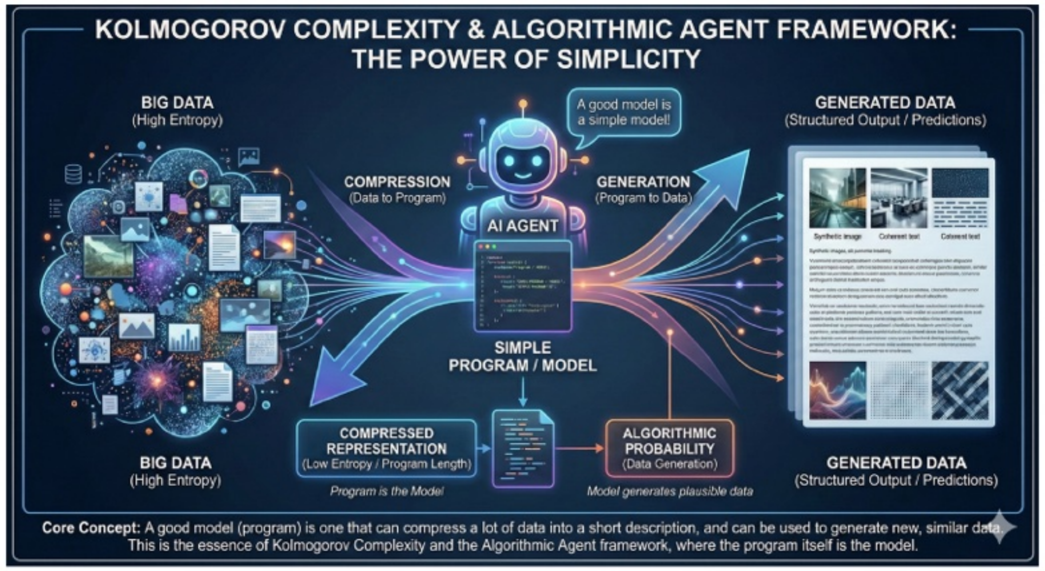
However, a fundamental theoretical limit exists: there is no general algorithm capable of determining the absolute shortest program to generate a specific dataset. This is a direct consequence of the halting problem in the theory of computation, which renders the calculation of ideal Kolmogorov complexity uncomputable.

Yet, while the search for a provably optimal program is formally undecidable, the scientific process remains viable. Progress is defined not by reaching the absolute limit, but by the iterative discovery of models with successively shorter description lengths—a process that is both possible and empirically verifiable.

The Algorithmic Weltanschauung Ruffini et al.

Modeling, Compression, Symmetry 28/83

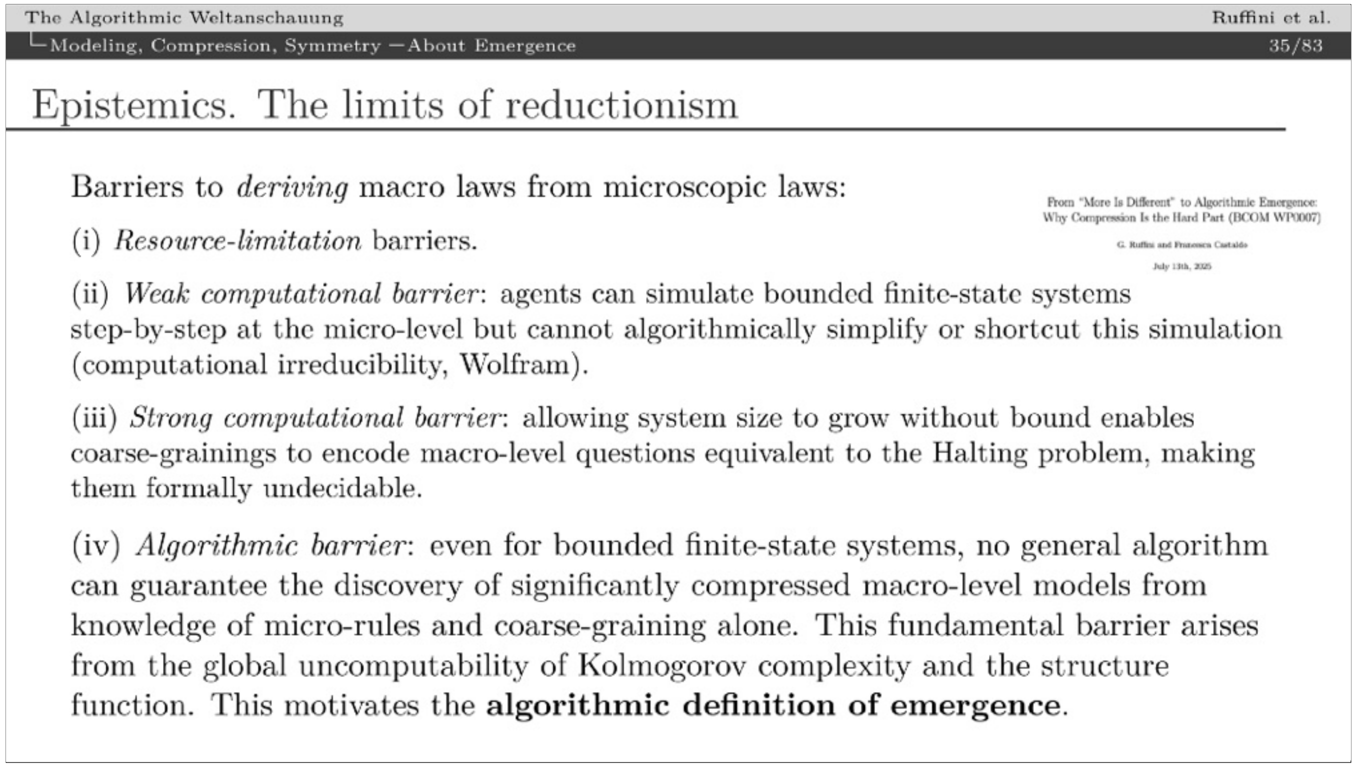
The power of simplicity


KOLMOGOROV COMPLEXITY & ALGORITHMIC AGENT FRAMEWORK: THE POWER OF SIMPLICITY
BIG DATA (High Entropy)
A good model is a simple model!
GENERATED DATA (Structured Output / Predictions)
COMPRESSION (Data to Program)
AI AGENT
GENERATION (Program to Data)
Synthetic image
Coherent text
Coherent text
SIMPLE PROGRAM / MODEL
COMPRESSED REPRESENTATION (Low Entropy / Program Length)
ALGORITHMIC PROBABILITY (Data Generation)
BIG DATA (High Entropy)
Program is the Model
Model generates plausible data
GENERATED DATA (Structured Output / Predictions)

Core Concept: A good model (program) is one that can compress a lot of data into a short description, and can be used to generate new, similar data. This is the essence of Kolmogorov Complexity and the Algorithmic Agent framework, where the program itself is the model.

In summary, scientific inquiry and modeling can be fundamentally understood as exercises in algorithmic compression. The objective is to identify concise programs that encode the complexity of the environment. These compressed representations are potent: they enable simulation and, crucially, exhibit robust generalization. By prioritizing short programs, agents avoid overfitting to local data and instead capture the true generative processes underlying reality. This drive for simplicity—exploiting regularities and patterns—is a universal principle governing agentic behavior, from simple organisms to the institutional operations of science.

However, a critical distinction exists between optimal truth and practical utility. Agents are not strictly compelled to find the absolute shortest or most precise description of the universe. Instead, they are incentivized to adopt models that are computationally efficient to discover and execute relative to their specific objective functions. A predator, for instance, relies on heuristic models for hunting and reproduction, not a derivation of the Standard Model. Nevertheless, the trajectory of intelligence suggests that the more capable agents—perhaps exemplified by Homo sapiens—eventually converge on these fundamental, deep models, as their superior predictive power yields the ultimate practical advantage.



Epistemics. The limits of reductionism

Barriers to *deriving* macro laws from microscopic laws:

From "More Is Different" to Algorithmic Emergence:
Why Compression Is the Hard Part (BCOM WP0007)

G. Ruffini and Francesca Castaldo

July 13th, 2025

(i) *Resource-limitation* barriers.

(ii) *Weak computational barrier*: agents can simulate bounded finite-state systems step-by-step at the micro-level but cannot algorithmically simplify or shortcut this simulation (computational irreducibility, Wolfram).

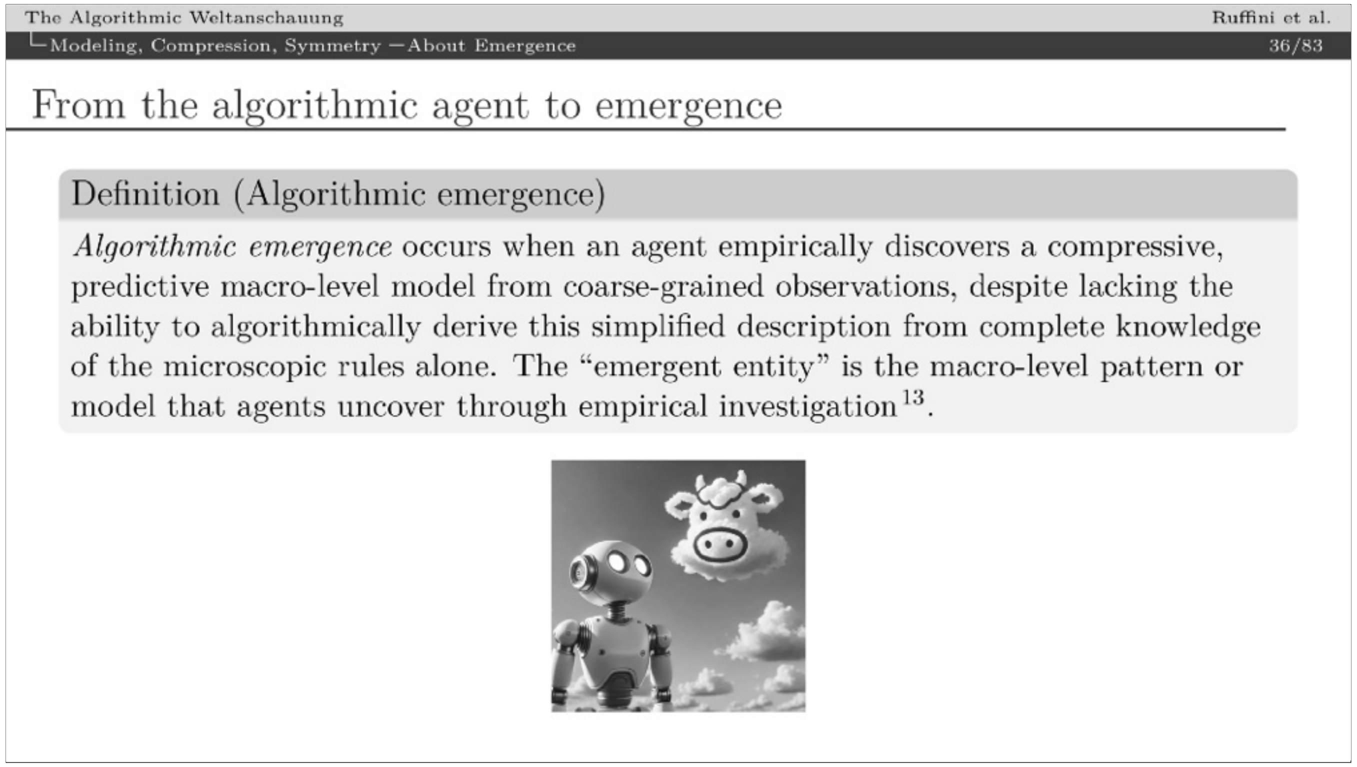
(iii) *Strong computational barrier*: allowing system size to grow without bound enables coarse-grainings to encode macro-level questions equivalent to the Halting problem, making them formally undecidable.

(iv) *Algorithmic barrier*: even for bounded finite-state systems, no general algorithm can guarantee the discovery of significantly compressed macro-level models from knowledge of micro-rules and coarse-graining alone. This fundamental barrier arises from the global uncomputability of Kolmogorov complexity and the structure function. This motivates the **algorithmic definition of emergence**.

Whatever the potential of deep reductionist models, their utility is strictly bounded by fundamental epistemic and computational barriers. Even when an agent successfully identifies valid microscopic laws, it faces the 'problem of reductionism': having the equations does not equate to having the solution.

The first barrier is physical: agents operate under finite constraints of time, memory, and energy, which limit the scope of computable predictions. Beyond resources, there are algorithmic barriers. In the 'weak' case (bounded systems), we encounter computational irreducibility, where knowledge of micro-laws yields no shortcut; the agent must simulate every intermediate step to predict the future. In the 'strong' case (unbounded systems), prediction becomes formally undecidable.

Furthermore, the inverse problem—systematically deriving efficient macro-laws solely from micro-laws and arbitrary coarse-graining—is strictly impossible in the general case. I call this barrier the 'Kolmogorov wall', a direct consequence of the halting problem. To circumvent this and successfully discover macro-laws, agents must identify specific coarse-grainings that preserve and respect the symmetries inherent in the micro-dynamics. This is highly non-trivial, but the success of physics and areas like statistical mechanics or condensed matter physics attest to its potential.



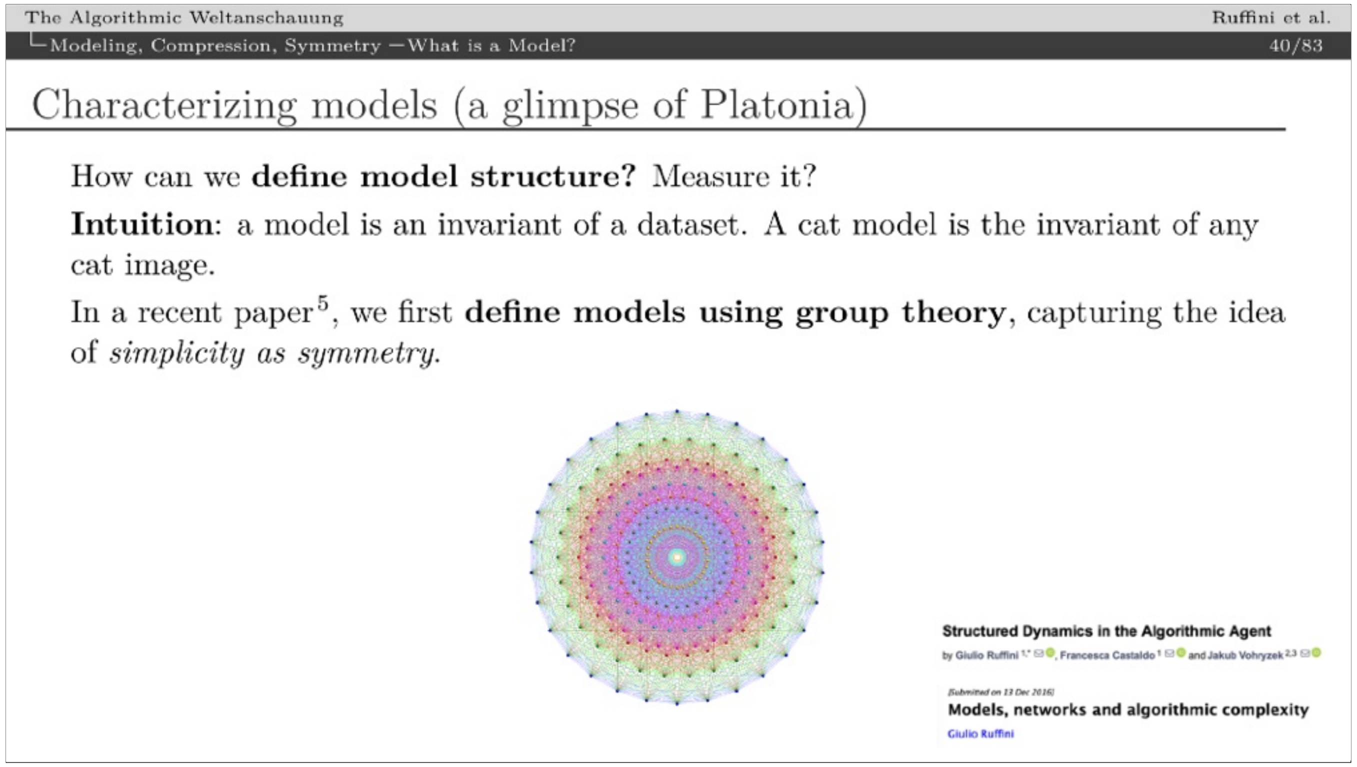
From the algorithmic agent to emergence

Definition (Algorithmic emergence)

Algorithmic emergence occurs when an agent empirically discovers a compressive, predictive macro-level model from coarse-grained observations, despite lacking the ability to algorithmically derive this simplified description from complete knowledge of the microscopic rules alone. The "emergent entity" is the macro-level pattern or model that agents uncover through empirical investigation[13].

These computational and epistemic barriers provide the rigorous foundation for an algorithmic theory of emergence. From the perspective of a bounded agent, emergence is defined operationally rather than ontologically: it occurs when the agent successfully identifies a predictive macro-law via coarse-graining.

Crucially, due to the aforementioned barriers (specifically the Kolmogorov wall), this emergent macro-theory is in general underivable from the microscopic laws alone. The computational cost to bridge the gap bottom-up can be prohibitive and formally unrealizable. Consequently, agents cannot only rely on deduction from first principles —valuable as they may be. Instead, the discovery of emergent laws calls for a distinct, complementary epistemic strategy: de novo compression. The agent must treat the macroscopic behavior as a fresh dataset, requiring independent empirical observation and compression efforts to identify patterns that are computationally inaccessible from the micro-scale.



Characterizing models (a glimpse of Platonia)

How can we **define model structure?** Measure it?

Intuition: a model is an invariant of a dataset. A cat model is the invariant of any cat image.

In a recent paper[5], we first **define models using group theory**, capturing the idea of *simplicity as symmetry.*

Structured Dynamics in the Algorithmic Agent

by Giulio Ruffini [1,*], Francesca Castaldo [1] and Jakub Vohryzek [2,3]

[Submitted on 13 Dec 2016]

Models, networks and algorithmic complexity

Giulio Ruffini

Ok, let's return to models. Fundamentally, a model is a program that compresses data. It achieves this by identifying the invariant properties of the dataset—the fixed rules from which the data can be generated step-by-step. Consider a dataset of images of a circle: rather than storing the coordinates of every point, we store the *invariant* equation (x-x0)^2 +(y-y0) ^2 =r^2. The 'program' is this static rule ("find solutions to this equation"), but the output is the full geometric shape. To generate a stack of images of circles we just need to specify a list of x0, y0 and r parameters and feed them to the program's core.

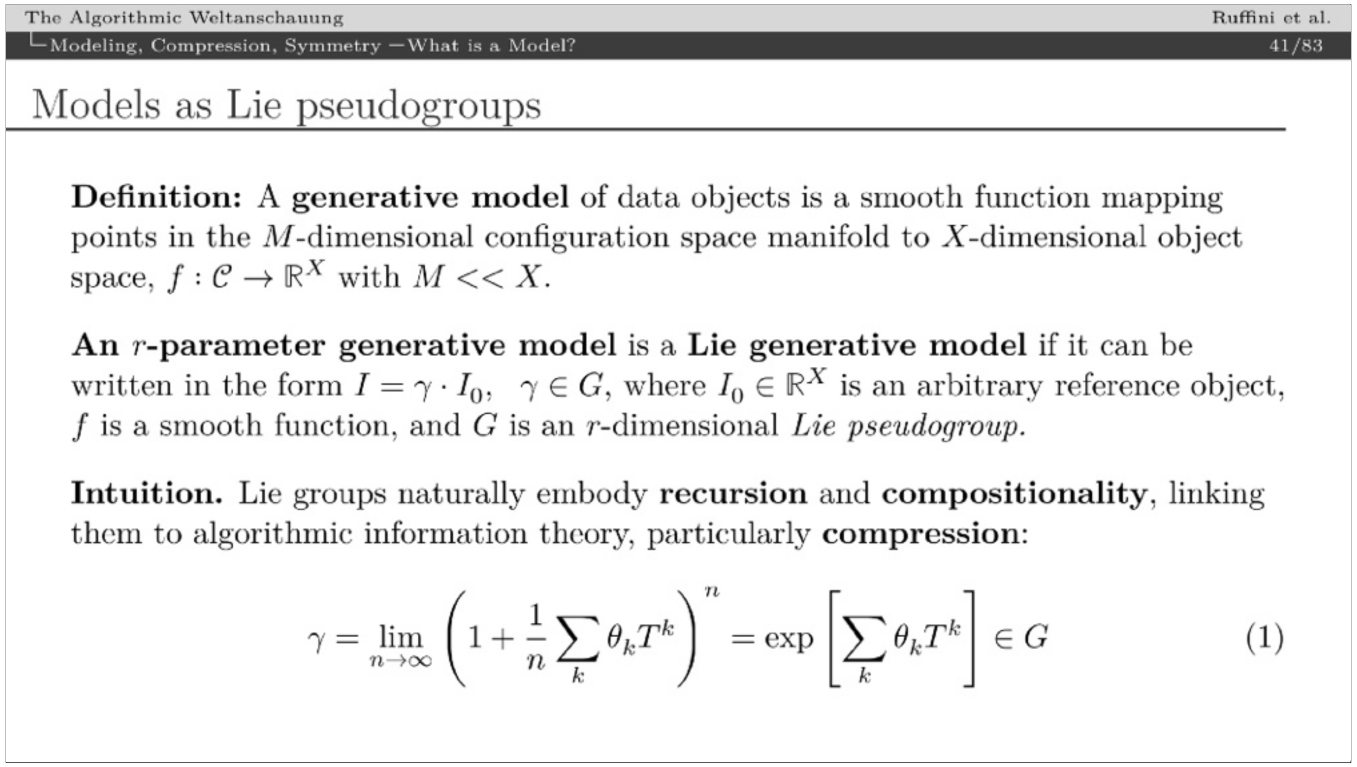
But how do we quantify this structure rigorously?

We can refine our definition using group theory, which links program length to symmetry. Specifically, the set of all transformations (such as translations, rotations or rescalings) that leave the dataset's (the stack of images) unchanged forms a Group of Invariances (G).

The ontology of "circle" is this dataset, or the short program that generates it (the model).

This provides a powerful metric: learning the model is formally equivalent to discovering this group. The 'structure' of the model is therefore quantified by the algorithmic complexity of its invariance group. A 'deep' model is one where a very short program (a simple group definition) can generate a vast, high-entropy dataset through recursive application of these symmetries.

Thus, we can refine the meaning of the model using group theory by linking the notions of program, compression, and symmetry.



The Algorithmic Weltanschauung — Ruffini et al.
Modeling, Compression, Symmetry — What is a Model? — 41/83

Models as Lie pseudogroups

Definition: A **generative model** of data objects is a smooth function mapping points in the M-dimensional configuration space manifold to X-dimensional object space, $f : \mathcal{C} \to \mathbb{R}^X$ with $M << X$.

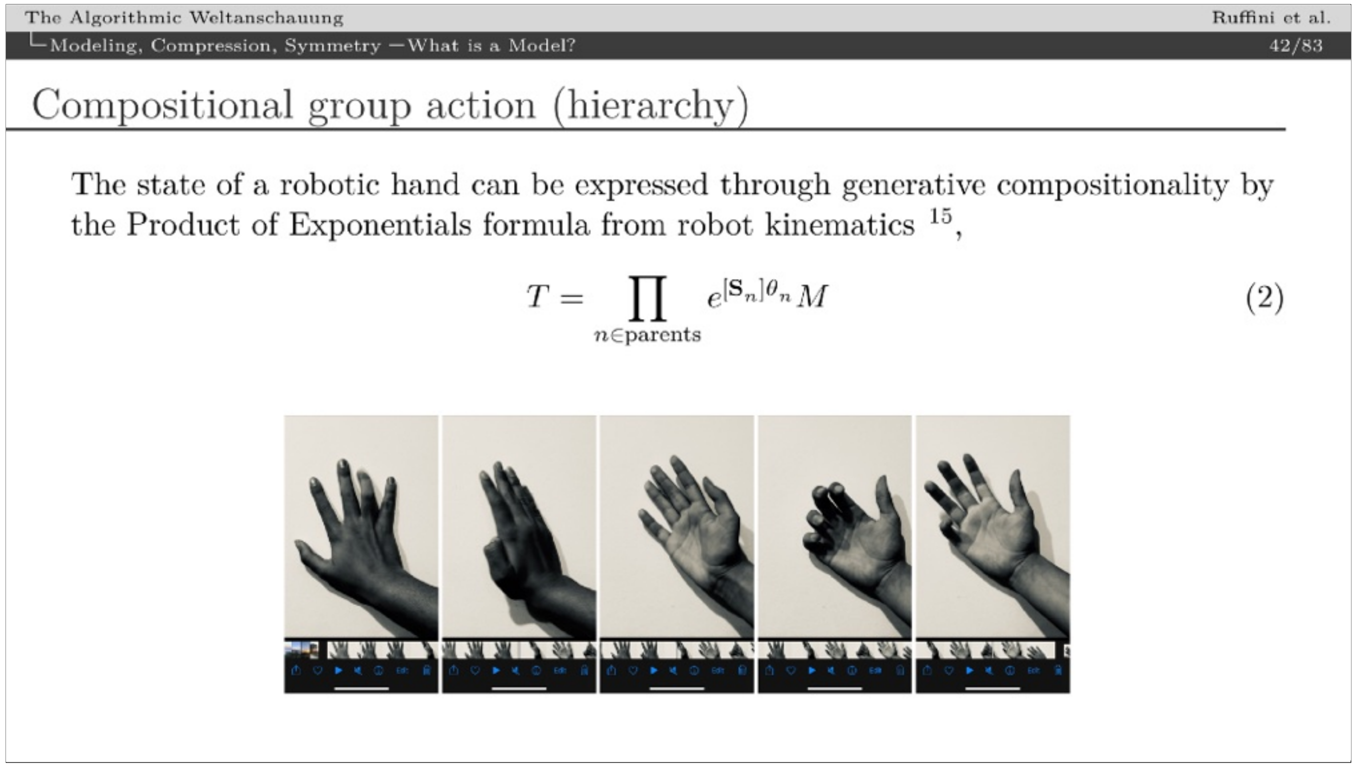
An r-parameter generative model is a **Lie generative model** if it can be written in the form $I = \gamma \cdot I_0, \quad \gamma \in G$, where $I_0 \in \mathbb{R}^X$ is an arbitrary reference object, f is a smooth function, and G is an r-dimensional *Lie pseudogroup.*

Intuition. Lie groups naturally embody **recursion** and **compositionality**, linking them to algorithmic information theory, particularly **compression**:

$$\gamma = \lim_{n\to\infty} \left(1 + \frac{1}{n} \sum_k \theta_k T^k\right)^n = \exp\left[\sum_k \theta_k T^k\right] \in G \tag{1}$$

To illustrate this, consider a generative model for a specific class of objects, such as hands or cats. Formally, this model is defined as a smooth map from a low-dimensional configuration manifold to a high-dimensional image space. By traversing trajectories within this configuration space, the model produces sequences of valid object variations.

We rigorously formalize this mechanism using Lie pseudogroups. In this framework, the Lie pseudogroup acts locally on a reference image (an archetype) using a compact parameter set to generate the full diversity of the dataset. Crucially, this generative action is inherently recursive and compositional—mirroring the fundamental structure of algorithmic programs (loops and nested functions). Consequently, the entire 'stack' of hand images is algorithmically equivalent to the short program—the 'model' of the hand—capable of generating it. Or, equivalently, to some Lie pseudogroup.



The Algorithmic Weltanschauung Ruffini et al.

└Modeling, Compression, Symmetry —What is a Model? 42/83

Compositional group action (hierarchy)

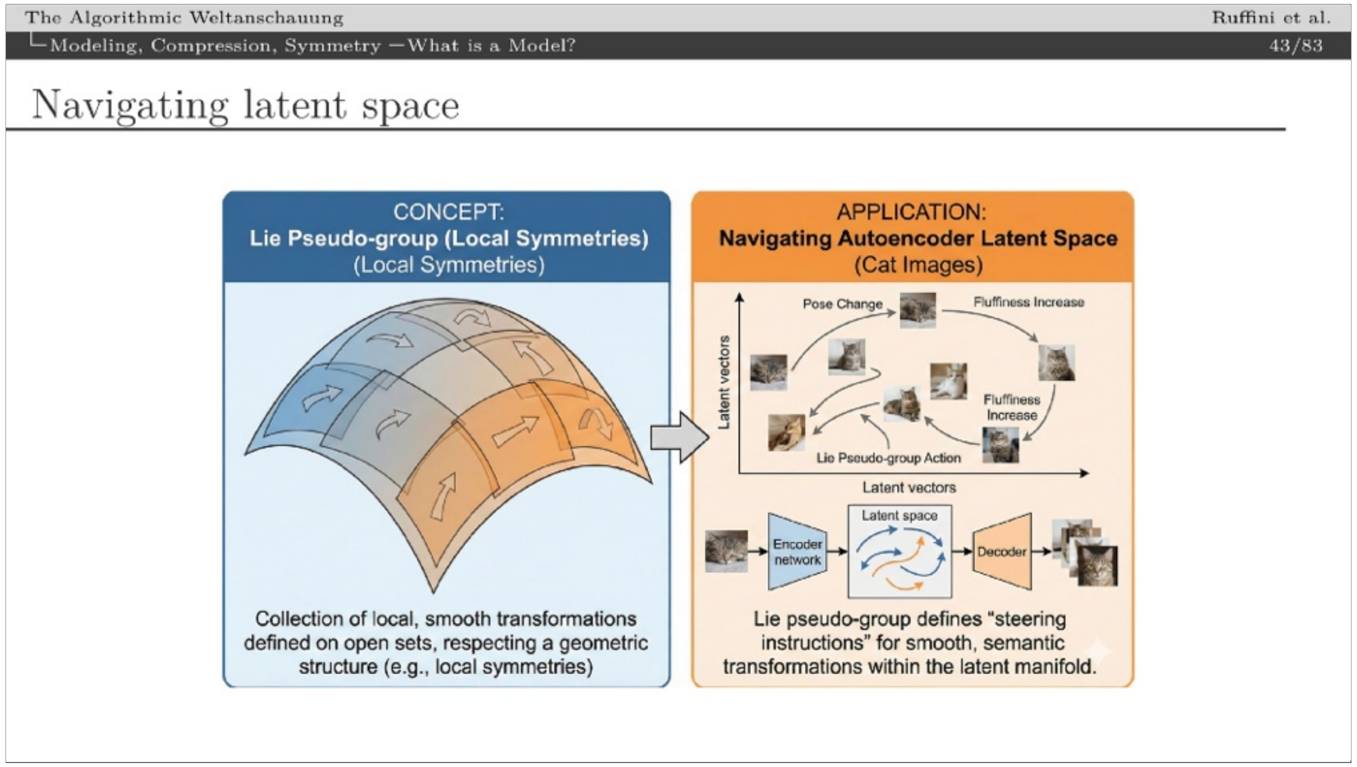
The state of a robotic hand can be expressed through generative compositionality by the Product of Exponentials formula from robot kinematics [15],

$$T = \prod_{n \in \text{parents}} e^{[\mathbf{S}_n]\theta_n} M \tag{2}$$

To make this concrete, consider the "Product of Exponentials" formula used in robot kinematics, as shown in the slide. Here, the final state of the hand is not retrieved from a database but computed via a chain of operations.

In this framework, M represents the "home" or reference configuration of the hand—the archetype. The exponential term represents the action of a specific joint (a local Lie group transformation) parameterized by an angle, θ.

Crucially, the product symbol embodies compositionality: the movement of a fingertip is the cumulative result of the wrist, knuckle, and finger joint transformations applied in sequence. This is a recursive, algorithmic process. By varying the small set of parameters in the configuration space, we can generate an effectively infinite manifold of hand images. The "model" is this compact kinematic equation, which serves as a highly compressed, generative program for the visual reality of "a hand."



Navigating latent space

CONCEPT:
Lie Pseudo-group (Local Symmetries)
(Local Symmetries)

Collection of local, smooth transformations defined on open sets, respecting a geometric structure (e.g., local symmetries)

APPLICATION:
Navigating Autoencoder Latent Space
(Cat Images)

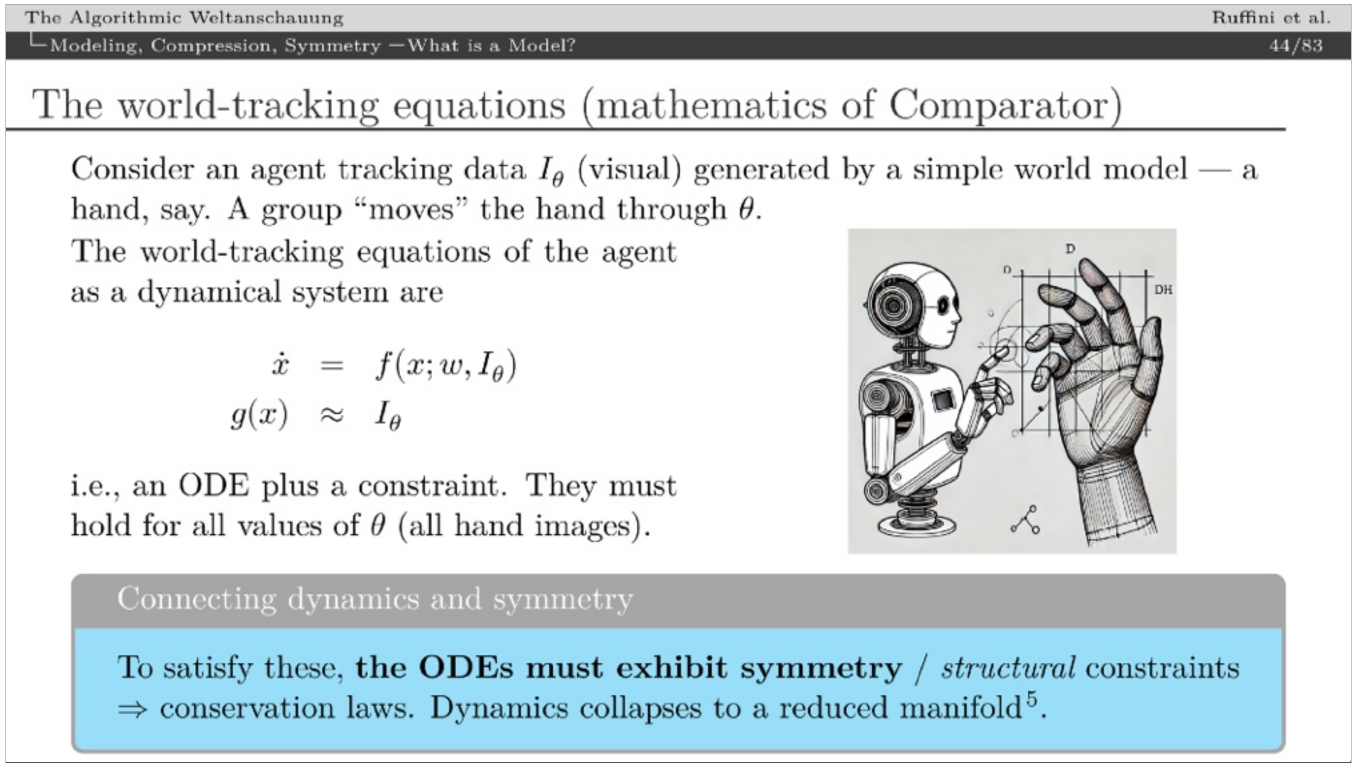
Lie pseudo-group defines “steering instructions” for smooth, semantic transformations within the latent manifold.

To generalize this further, imagine we have trained a compressive autoencoder on billions of cat images. Once trained, this model defines a latent space—a compressed, high-dimensional manifold where every point corresponds to a valid image of a cat.

In this system, the "generative model" is not just a static database, but a mechanism for navigating this latent space. Specific directions or dimensions within this space correspond to semantic changes in the object, such as determining orientation, translation, pose, or even hair type (e.g., "fluffiness increase").

We formalize this navigation using the concept of Lie Pseudo-groups.

Unlike global symmetries that apply everywhere (like rotating a sphere), a Lie Pseudo-group describes a collection of local smooth transformations defined on patches of the manifold. This mathematical structure defines the "steering instructions" for the model: it dictates how to move smoothly from one point in the latent space to another to transform the image (e.g., changing the cat's pose) while ensuring the result remains valid within the geometric structure of the data.



The Algorithmic Weltanschauung Ruffini et al.

└Modeling, Compression, Symmetry —What is a Model? 44/83

The world-tracking equations (mathematics of Comparator)

Consider an agent tracking data I_θ (visual) generated by a simple world model — a hand, say. A group "moves" the hand through θ.

The world-tracking equations of the agent as a dynamical system are

$$\begin{aligned} \dot{x} &= f(x; w, I_\theta) \\ g(x) &\approx I_\theta \end{aligned}$$

i.e., an ODE plus a constraint. They must hold for all values of θ (all hand images).

Connecting dynamics and symmetry

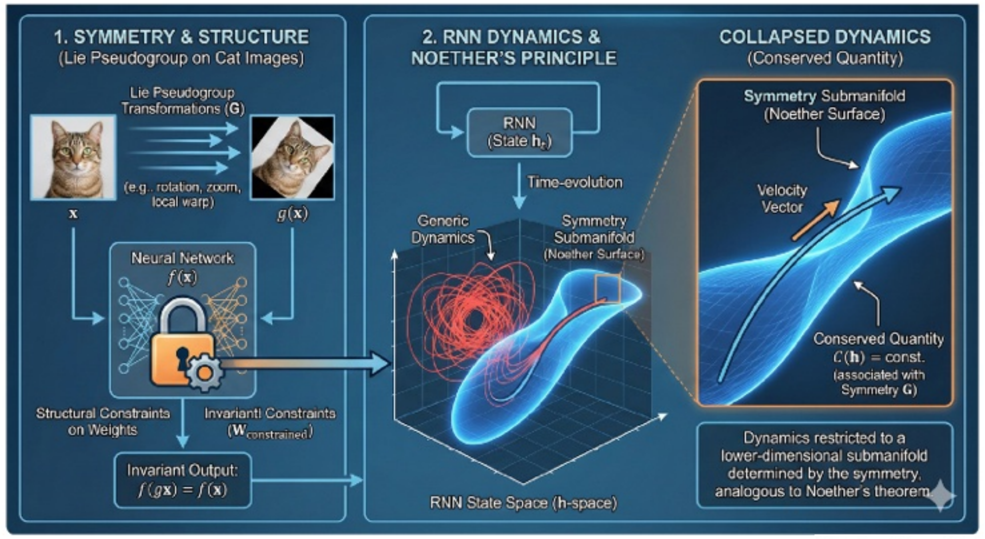
To satisfy these, **the ODEs must exhibit symmetry** / *structural* constraints ⇒ conservation laws. Dynamics collapses to a reduced manifold[5].

We can now generalize this into a central hypothesis: all structured data received by agents is fundamentally generated via Lie groups. This posits a theoretical equivalence: that algorithmic programs—and potentially Turing machines themselves—can be expressed using the machinery of Lie pseudogroups. We adopt this as our working premise: world data is compositionally and recursively generated by Lie symmetries.

Consider an agent, modeled as a dynamical system, tasked with tracking this data. Tracking is defined operationally: the agent must successfully match the incoming data stream using its internal generative model.

This interaction leads to a rigorous dynamical constraint. As shown in the world-tracking equations, for an agent to successfully synchronize with a world generated by symmetries, its internal dynamics must mirror those symmetries. This structural alignment enforces conservation laws within the agent, compelling its state space to collapse onto a reduced manifold—a low-dimensional attractor that embodies the structure of the world it observes.

From Symmetry to Dynamics


1. SYMMETRY & STRUCTURE
(Lie Pseudogroup on Cat Images)
Lie Pseudogroup Transformations (**G**)
(e.g., rotation, zoom, local warp)
$\mathbf{x}$
$g(\mathbf{x})$
Neural Network $f(\mathbf{x})$
Structural Constraints on Weights
Invarianti Constraints ($\mathbf{W}_{\text{constrained}}$)
Invariant Output: $f(g\mathbf{x}) = f(\mathbf{x})$
2. RNN DYNAMICS & NOETHER'S PRINCIPLE
RNN (State $\mathbf{h}_t$)
Time-evolution
Generic Dynamics
Symmetry Submanifold (Noether Surface)
RNN State Space (h-space)
COLLAPSED DYNAMICS
(Conserved Quantity)
Symmetry Submanifold (Noether Surface)
Velocity Vector
Conserved Quantity $C(\mathbf{h})$ = const. (associated with Symmetry **G**)
Dynamics restricted to a lower-dimensional submanifold determined by the symmetry, analogous to Noether's theorem

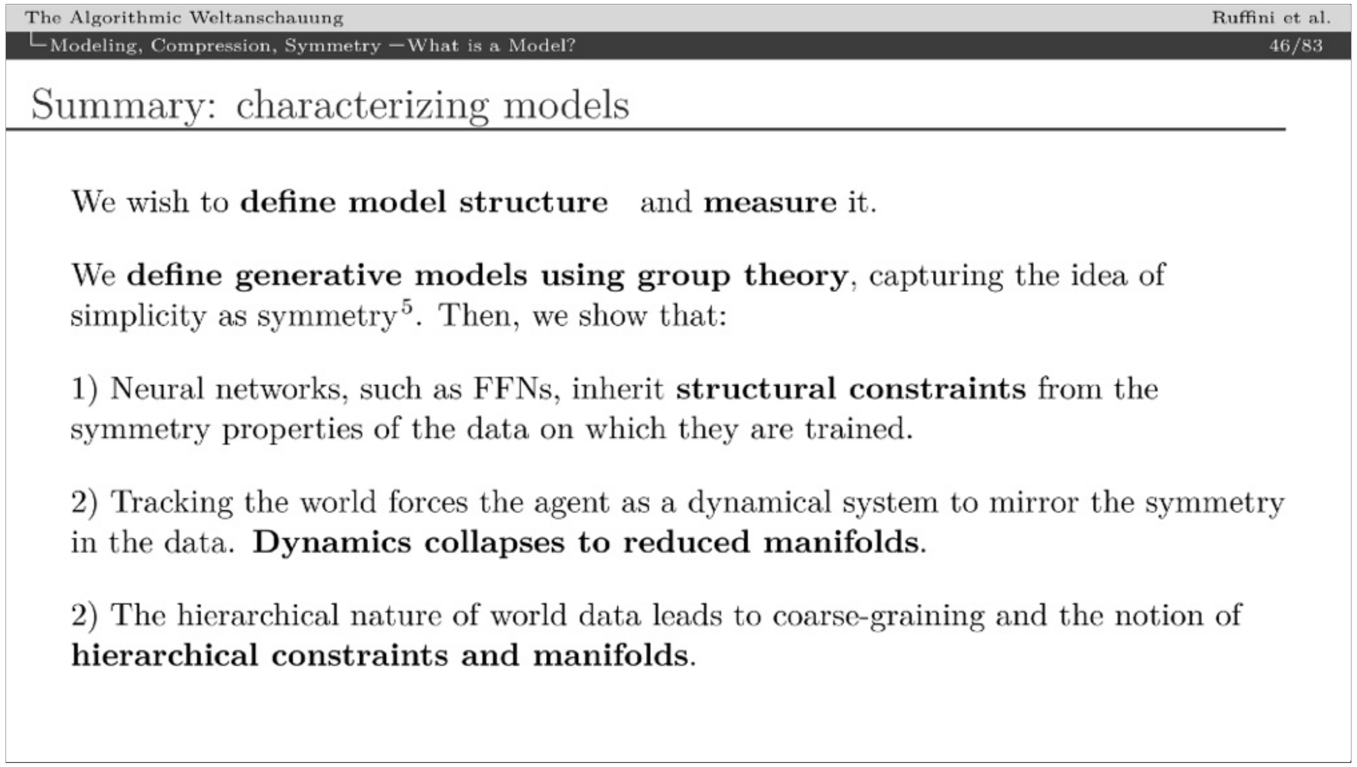
Structured Dynamics in the Algorithmic Agent

by Giulio Ruffini [1,*], Francesca Castaldo [1] and Jakub Vohryzek

We can draw a direct parallel to physics via Noether's principle. In physical systems, continuous symmetries (like rotation) enforce conservation laws (like angular momentum), which in turn constrain dynamics to lower-dimensional surfaces (e.g., planetary orbits confined to a plane).

Similarly, for an algorithmic agent, the necessity of tracking a Lie-generated world imposes structural constraints on its internal connectivity.

These constraints manifest as algorithmic conservation laws within the neural dynamics, effectively collapsing the system's vast potential state space onto a low-dimensional symmetry submanifold. Thus, despite possessing billions of degrees of freedom (neurons), the agent's actual trajectories are strictly confined to the geometry dictated by the world's underlying Lie structure.



Summary: characterizing models

We wish to **define model structure** and **measure** it.

We **define generative models using group theory**, capturing the idea of simplicity as symmetry[5]. Then, we show that:

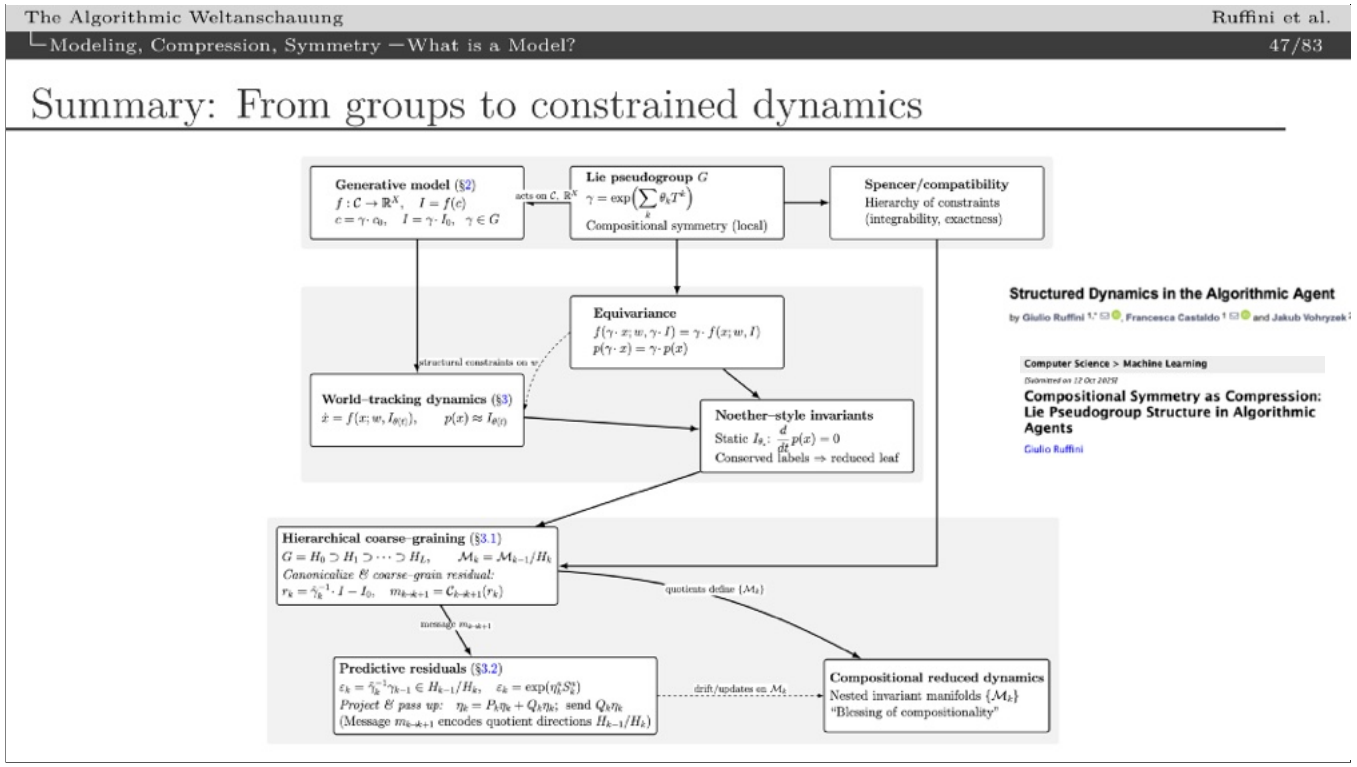
1) Neural networks, such as FFNs, inherit **structural constraints** from the symmetry properties of the data on which they are trained.

2) Tracking the world forces the agent as a dynamical system to mirror the symmetry in the data. **Dynamics collapses to reduced manifolds**.

2) The hierarchical nature of world data leads to coarse-graining and the notion of **hierarchical constraints and manifolds**.

In summary, we propose a rigorous characterization of model structure based on Lie pseudogroups. By defining generative models through the lens of group theory, we capture the essence of algorithmic simplicity as symmetry. This framework yields three fundamental insights:

Structural Inheritance: Neural networks trained on such data do not merely learn patterns; they inherit structural constraints directly from the symmetry properties of the data.
Dynamical Collapse: The requirement to track a Lie-generated world forces the agent as a dynamical system to mirror these symmetries. This structural alignment causes the system's trajectory to collapse onto reduced manifolds—low-dimensional subspaces within the high-dimensional state space.
Hierarchical Architecture: Finally, since world data is inherently compositional, this process naturally extends to multiple scales. The Lie group structure necessitates coarse-graining, leading to a nested architecture of hierarchical constraints and manifolds, where high-level abstractions systematically constrain lower-level dynamics.



The Algorithmic Weltanschauung — Ruffini et al.

└Modeling, Compression, Symmetry —What is a Model? — 47/83

Summary: From groups to constrained dynamics

Generative model (§2)
$f : \mathcal{C} \to \mathbb{R}^X, \quad I = f(c)$
$c = \gamma \cdot c_0, \quad I = \gamma \cdot I_0, \quad \gamma \in G$

Lie pseudogroup G
$\gamma = \exp\left(\sum_k \theta_k T^k\right)$
Compositional symmetry (local)

Spencer/compatibility
Hierarchy of constraints
(integrability, exactness)

Equivariance
$f(\gamma \cdot x; w, \gamma \cdot I) = \gamma \cdot f(x; w, I)$
$p(\gamma \cdot x) = \gamma \cdot p(x)$

World–tracking dynamics (§3)
$\dot{x} = f(x; w, I_{\theta(t)}), \qquad p(x) \approx I_{\theta(t)}$

Noether–style invariants
Static I_{θ_*}: $\frac{d}{dt} p(x) = 0$
Conserved labels ⇒ reduced leaf

Hierarchical coarse–graining (§3.1)
$G = H_0 \supset H_1 \supset \cdots \supset H_L, \qquad \mathcal{M}_k = \mathcal{M}_{k-1}/H_k$
Canonicalize & coarse–grain residual:
$r_k = \hat{\gamma}_k^{-1} \cdot I - I_0, \quad m_{k \to k+1} = \mathcal{C}_{k \to k+1}(r_k)$

Predictive residuals (§3.2)
$\varepsilon_k = \hat{\gamma}_k^{-1} \gamma_{k-1} \in H_{k-1}/H_k, \quad \varepsilon_k = \exp(\eta_k^a S_k^a)$
Project & pass up: $\eta_k = P_k \eta_k + Q_k \eta_k$; send $Q_k \eta_k$
(Message $m_{k \to k+1}$ encodes quotient directions H_{k-1}/H_k)

Compositional reduced dynamics
Nested invariant manifolds $\{\mathcal{M}_k\}$
"Blessing of compositionality"

acts on $\mathcal{C}$, $\mathbb{R}^X$
structural constraints on w
quotients define $\{\mathcal{M}_k\}$
message $m_{k \to k+1}$
drift/updates on $\mathcal{M}_k$

Structured Dynamics in the Algorithmic Agent

by Giulio Ruffini [1,*], Francesca Castaldo [1] and Jakub Vohryzek

Computer Science > Machine Learning

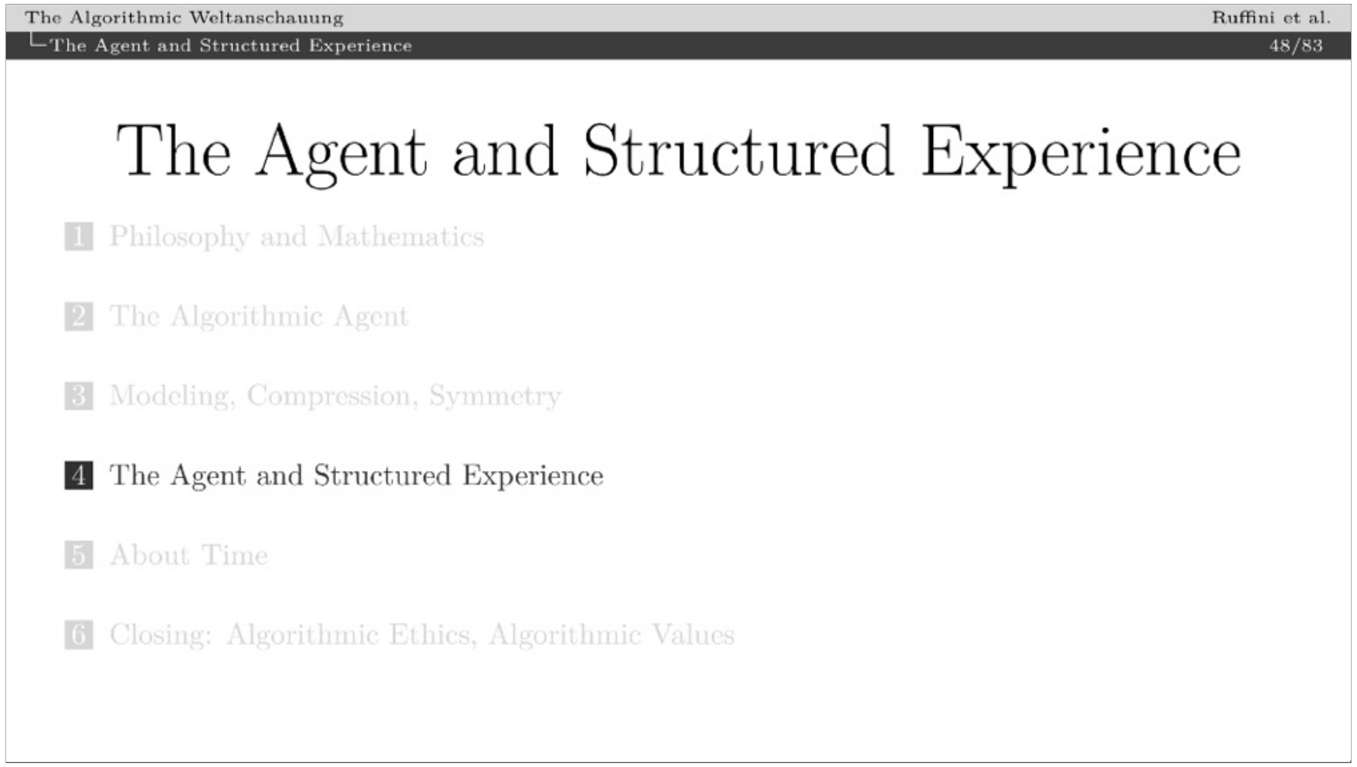
[Submitted on 12 Oct 2025]

Compositional Symmetry as Compression: Lie Pseudogroup Structure in Algorithmic Agents

Giulio Ruffini

And here is a schematic summary of this argumentation, with a bit more mathematical detail on how Lie pseudogroups link data generation, agent structure, and dynamical collapse into hierarchical reduced manifolds.

You can find details in the preprints.



The Agent and Structured Experience

1 Philosophy and Mathematics

2 The Algorithmic Agent

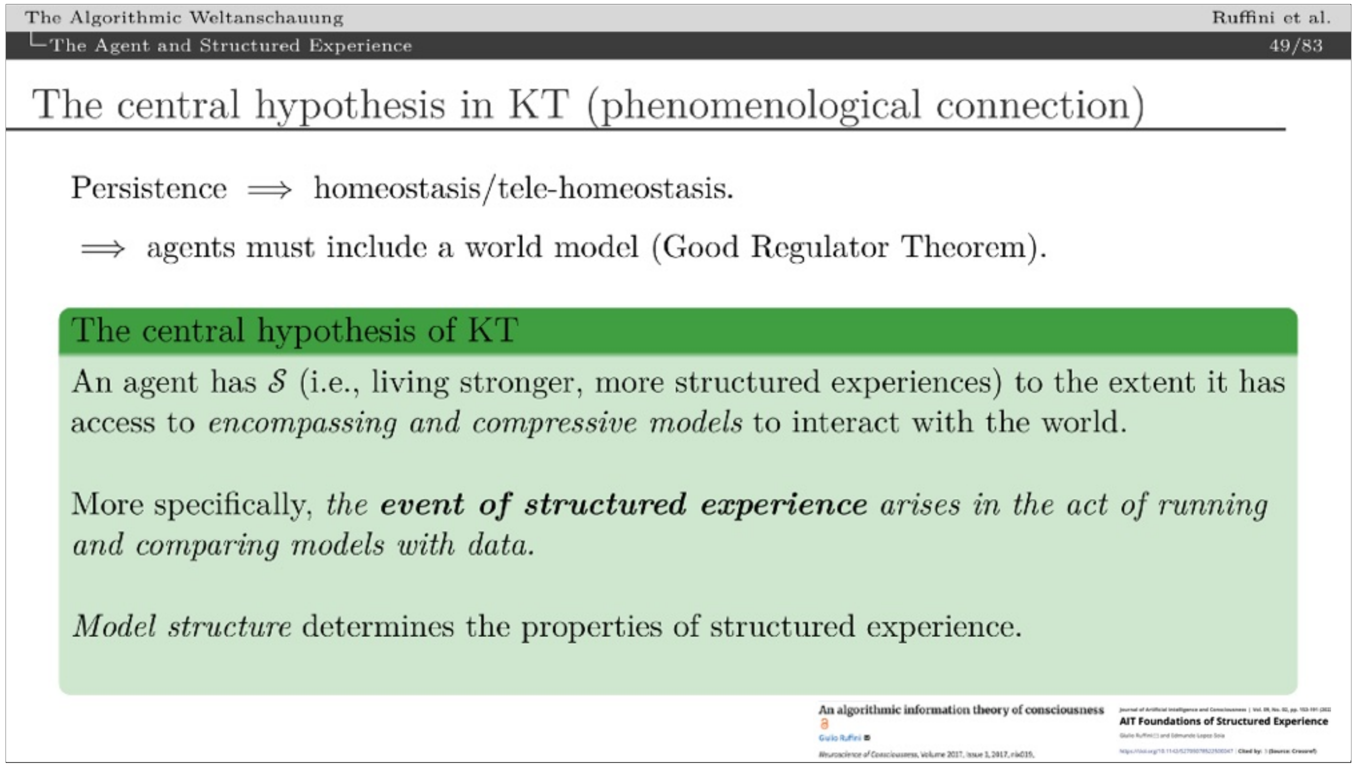
3 Modeling, Compression, Symmetry

4 The Agent and Structured Experience

5 About Time

6 Closing: Algorithmic Ethics, Algorithmic Values

We now have the tools to discuss how the algorithmic agent mathematics link with *subjective structured experience.*



The central hypothesis in KT (phenomenological connection)

Persistence $\implies$ homeostasis/tele-homeostasis.

$\implies$ agents must include a world model (Good Regulator Theorem).

The central hypothesis of KT

An agent has $\mathcal{S}$ (i.e., living stronger, more structured experiences) to the extent it has access to *encompassing and compressive models* to interact with the world.

More specifically, *the* ***event of structured experience*** *arises in the act of running and comparing models with data.*

Model structure determines the properties of structured experience.

An algorithmic information theory of consciousness
Giulio Ruffini
Neuroscience of Consciousness, Volume 2017, Issue 1, 2017, nix019,

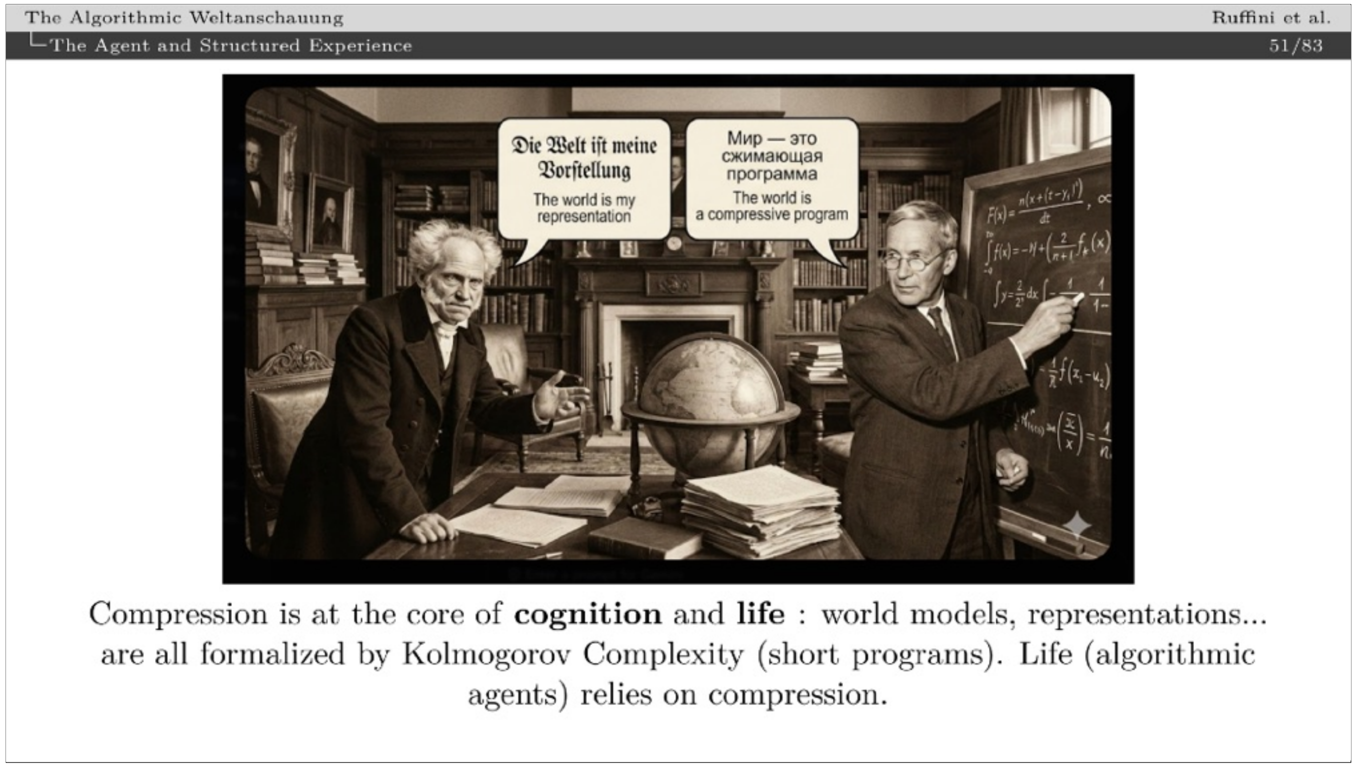
Journal of Artificial Intelligence and Consciousness | Vol. 09, No. 02, pp. 153-191 (202
AIT Foundations of Structured Experience
Giulio Ruffini and Edmundo Lopez Sola
https://doi.org/10.1142/S2705078522500047 | Cited by: 1 (Source: Crossref)

We now turn to the connection with first-person experience.

We grounded our framework in the fundamental principle of algorithmic persistence. To persist, an agent must maintain homeostasis (and tele-homeostasis), a requirement that, under the Good Regulator Theorem, necessitates the internalization of a world model.

This leads to the Central Hypothesis of Kolmogorov Theory (KT) regarding phenomenology. We posit that while 'pure' experience may be a primitive baseline, structured experience (S) emerges strictly to the extent that an agent utilizes encompassing and compressive models to interact with reality. Specifically, the event of structured experience arises dynamically in the act of running and comparing these models with incoming data.

Crucially, this implies that the **structure** of the model determines the **structure** of the experience. By characterizing model structure through Lie pseudogroups and reduced manifolds (as established in our previous sections), we provide a concrete mathematical basis for this hypothesis: the symmetries of the agent's internal generative models directly sculpt the geometry of its subjective experience.

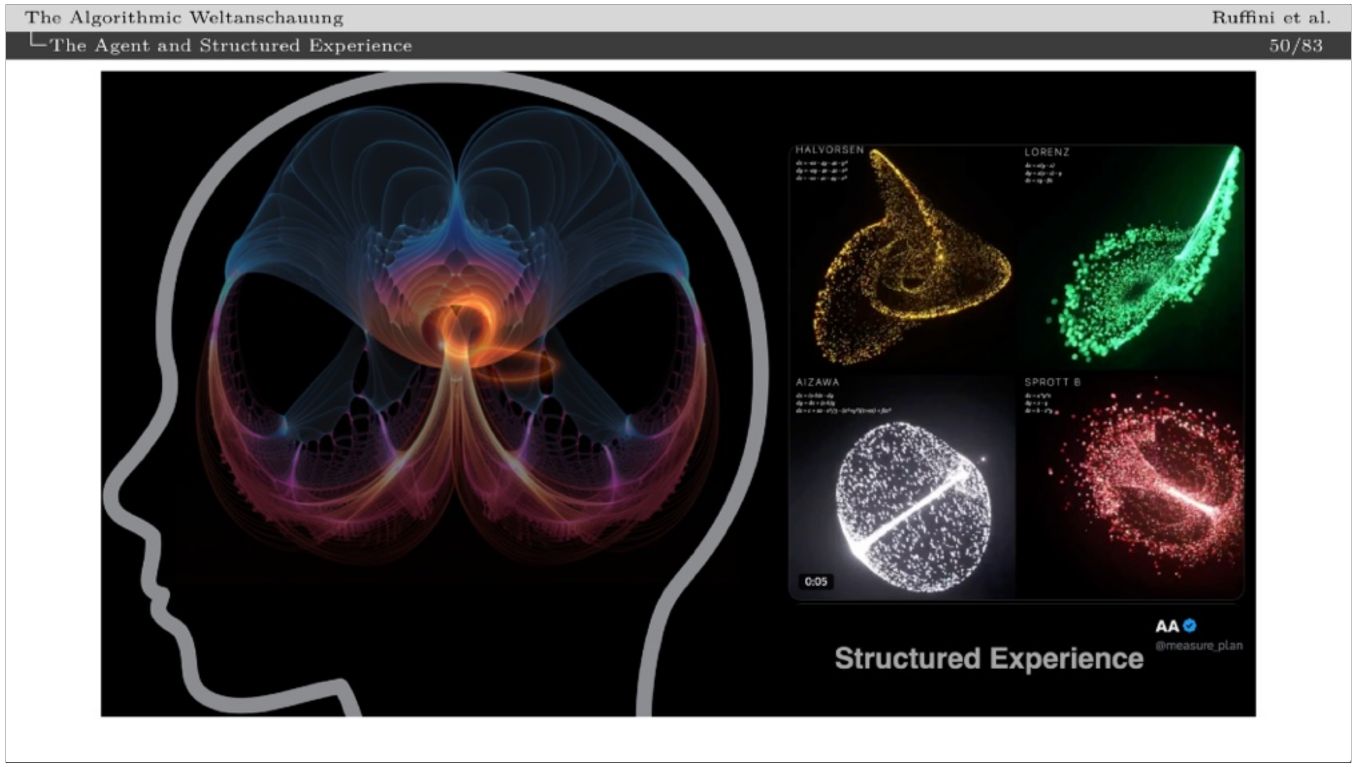

Die Welt iſt meine Vorſtellung
The world is my representation
Мир — это сжимающая программа
The world is a compressive program

Compression is at the core of **cognition** and **life** : world models, representations... are all formalized by Kolmogorov Complexity (short programs). Life (algorithmic agents) relies on compression.

One can say we are rephrasing old ideas into algorithmic terms.

For example, from Schopenhauer's "The world is my representation" (my perception of what the world is is a "representation" or model) to how Kolmogorov may have stated it: "my perception of the world is a compressive program.

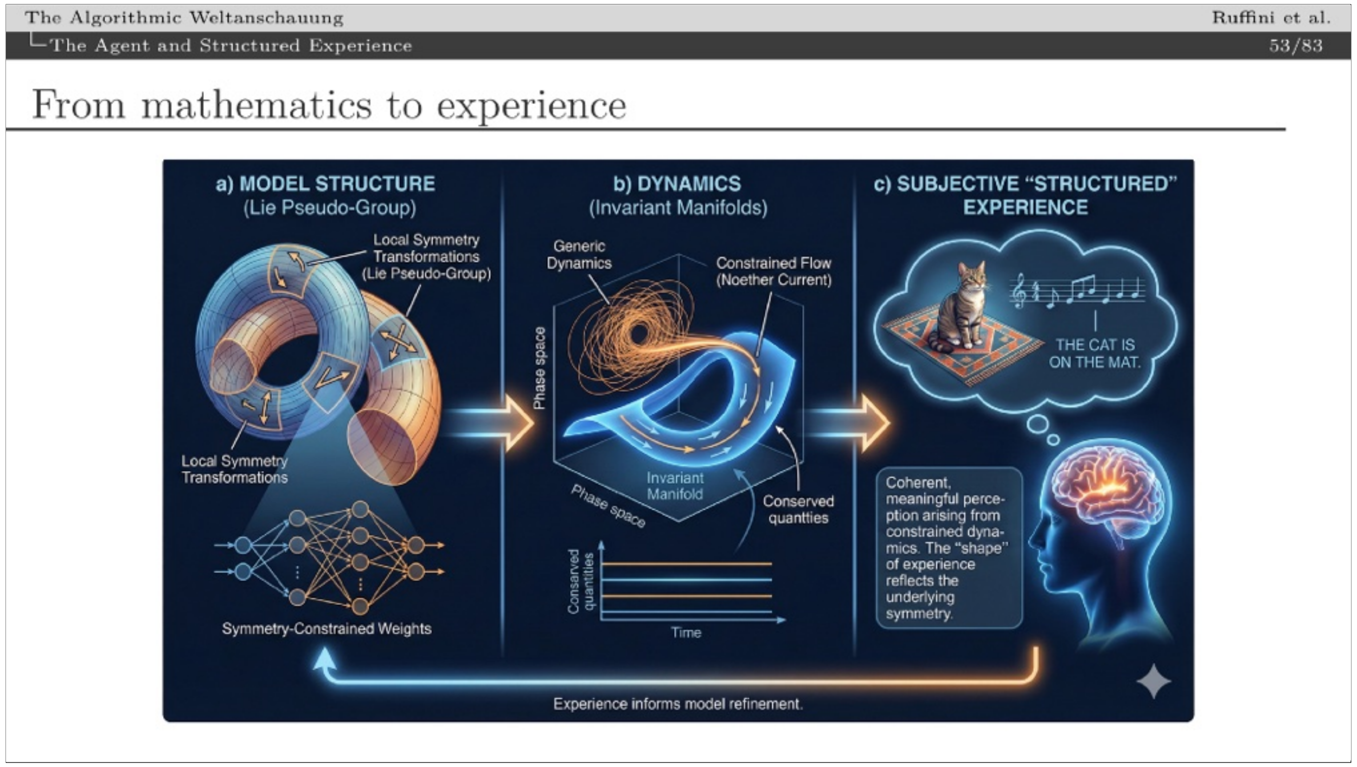
Well, this is how I would state it!



This proposition carries profound implications, moving beyond abstract theory to concrete prediction. It posits that structured experience is fundamentally shaped by the structural constraints of agent programs—instantiated physically as neural connectivity in the case of humans.

This mapping manifests dynamically: the static structure of the agent's program governs the geometry and topology of its neural reduced manifolds. We argue that these specific dynamical shapes—the attractors and trajectories within the agent's state space—are the direct physical correlates of structured experience.

This constitutes an algorithmic and dynamical reformulation of the Neural Correlates of Consciousness (NCC). Yet, we propose a relationship stronger than mere correlation. This framework implies causality: if you alter the dynamical structure—whether through meditation, pharmacological intervention, or brain stimulation—you necessarily alter the topology of the manifold, and thus the structure of the experience itself. Ultimately, we argue for an identity relation: the geometry of the dynamical attractor is the structure of the experience.



From mathematics to experience

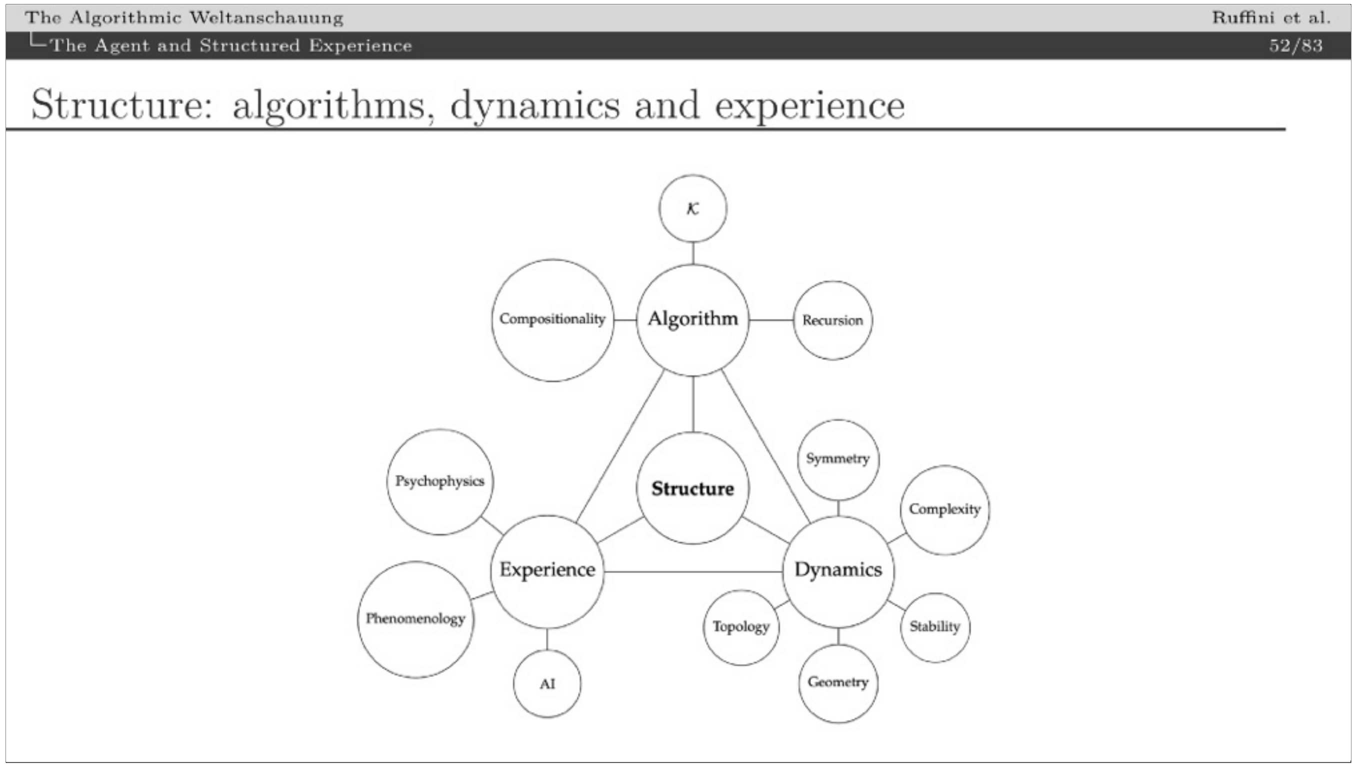
a) MODEL STRUCTURE
(Lie Pseudo-Group)
Local Symmetry Transformations (Lie Pseudo-Group)
Local Symmetry Transformations
Symmetry-Constrained Weights
b) DYNAMICS
(Invariant Manifolds)
Generic Dynamics
Constrained Flow (Noether Current)
Phase space
Phase space
Invariant Manifold
Conserved quantties
Conserved quantities
Time
c) SUBJECTIVE "STRUCTURED" EXPERIENCE
THE CAT IS ON THE MAT.
Coherent, meaningful perception arising from constrained dynamics. The "shape" of experience reflects the underlying symmetry.
Experience informs model refinement.

This brings us to our final, unified conclusion: Model Structure, Dynamics, and Subjective Experience are not separate domains, but intrinsically linked phases of a single process.

Inheritance: First, Model Structure—formalized here as Lie pseudogroups—is not arbitrary. It inherits its architecture directly from the fundamental symmetries of the external world.

Constraint: Second, this inherited structure acts as a rigorous dynamical constraint. It forces the agent's neural activity to collapse from high-dimensional possibilities into specific reduced manifolds. The agent's internal state is physically compelled to flow along these symmetry-defined paths.

Identity: Finally, we propose that this dynamical geometry is the mathematical substrate of phenomenology. The specific topology of these reduced manifolds—the 'shape' of the agent's constrained dynamics—does not merely correlate with perception. It shapes, and indeed is, the structured experience itself.



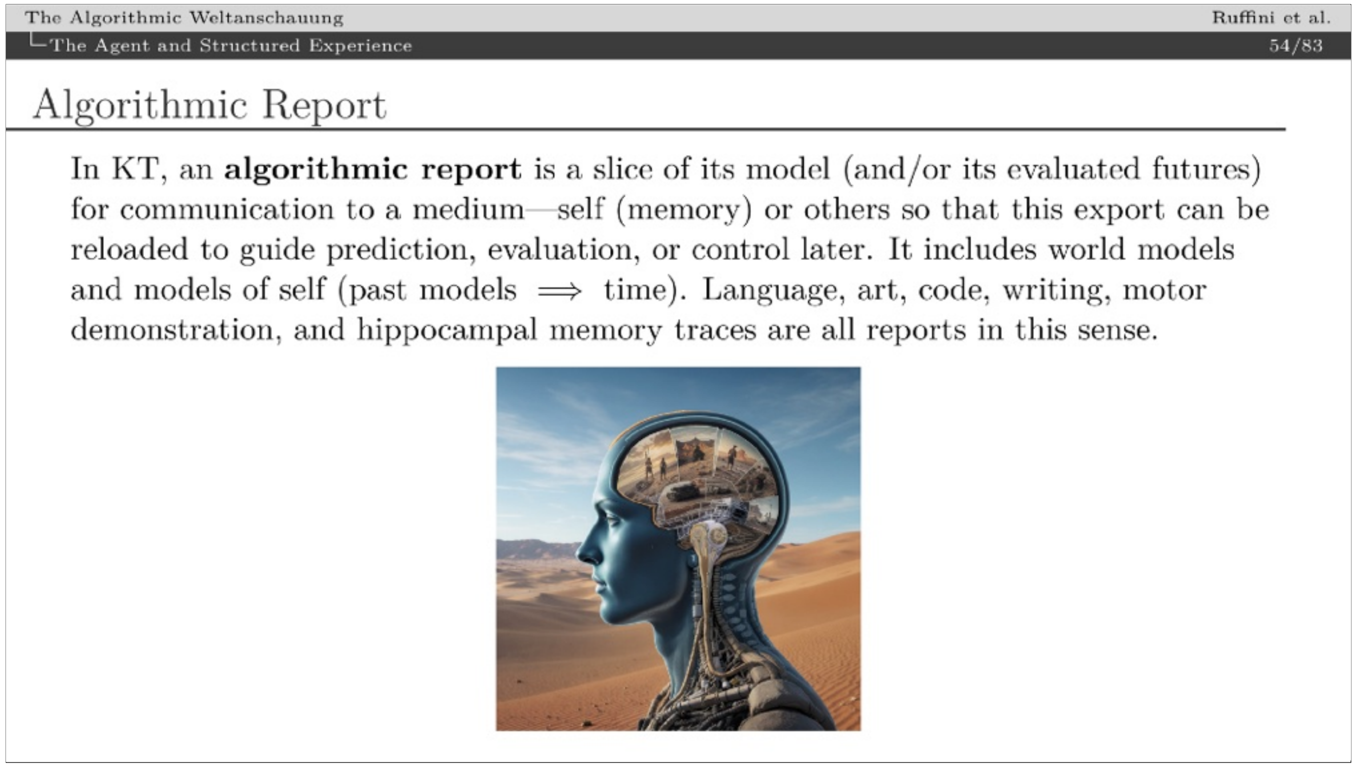
Structure: algorithms, dynamics and experience

The epicenter of this theoretical framework is the mathematical concept of Structure. It serves as the unifying nexus connecting three distinct domains:

Algorithmic Information Theory: Here, structure is rigorously operationalized through recursion, compositionality, and Kolmogorov complexity (K)—the fundamental metrics of program length and compressibility.

Dynamics: In the physical substrate, this algorithmic structure manifests as symmetry. This imposes strict constraints on the system, determining the geometry, topology, and stability of the resulting dynamical attractors.

Experience: Finally, these foundations underpin Phenomenology. By linking subjective experience to these rigorous mathematical definitions, we establish a framework not only for quantifying biological consciousness but also for evaluating the potential for structured experience in Artificial Intelligence.



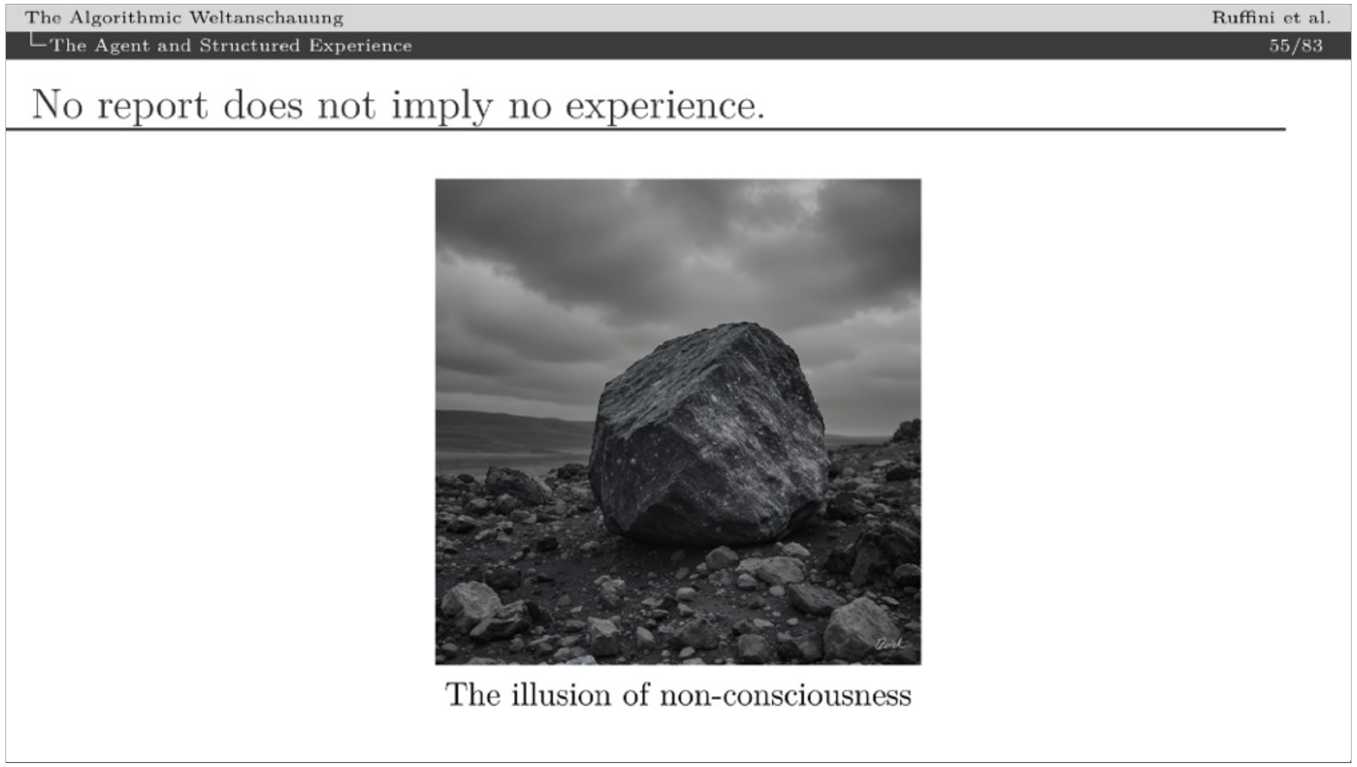
Algorithmic Report

In KT, an **algorithmic report** is a slice of its model (and/or its evaluated futures) for communication to a medium—self (memory) or others so that this export can be reloaded to guide prediction, evaluation, or control later. It includes world models and models of self (past models $\implies$ time). Language, art, code, writing, motor demonstration, and hippocampal memory traces are all reports in this sense.

Finally, we must distinguish between the existence of structured experience and the capacity to report it.

In KT, we define an algorithmic report not as the experience itself, but as a specific export operation: it is a compressed 'slice' or projection of the agent's active world-model. This slice is transmitted to a medium—either internally to the 'self' (via hippocampal memory traces) or externally to others (via language, art, or code).

Crucially, the absence of a report does not prove the absence of experience. A system may possess a rich, structured internal model but lack the capacity to export it due to communication blocks (e.g., Locked-In Syndrome) or memory deficits. Thus, the report is a functional tool: it exists so that a model state can be reloaded later to guide prediction and control, distinct from the immediate reality of the experience.



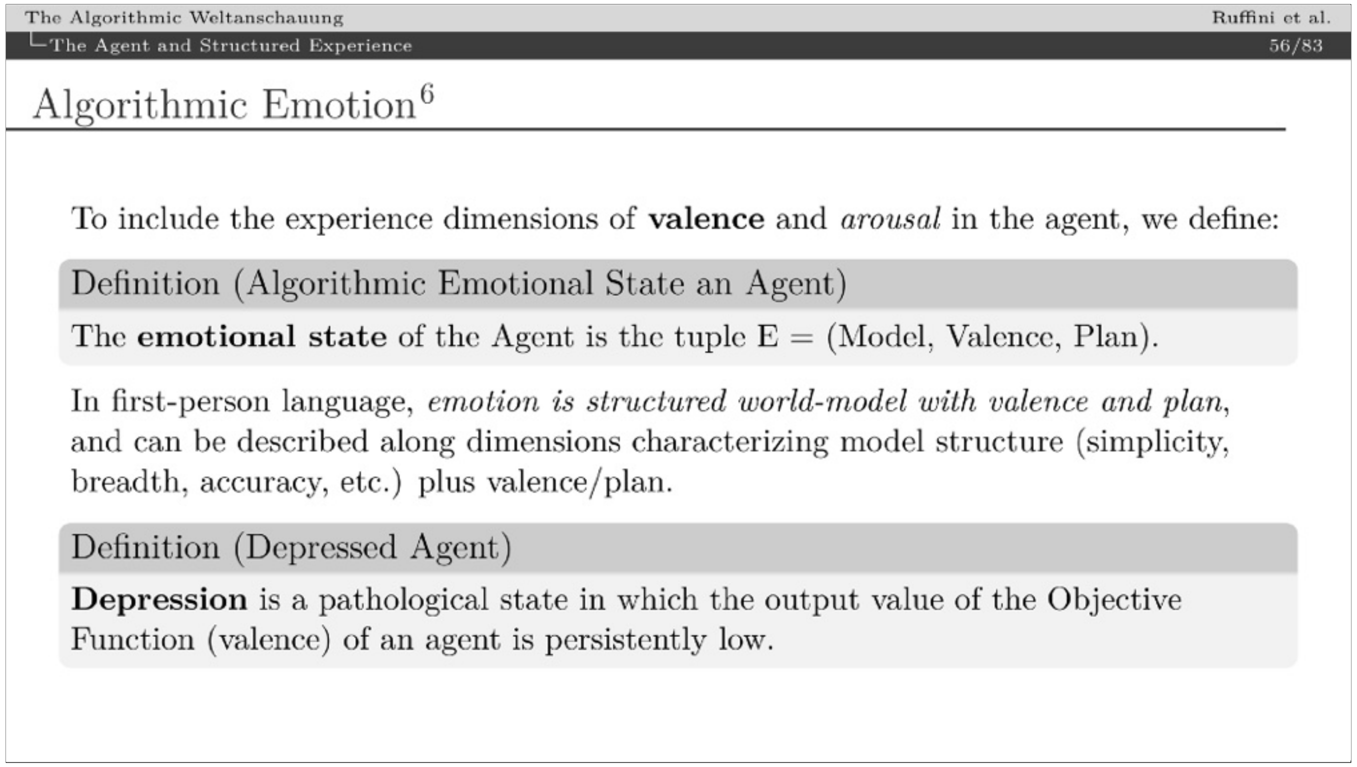
No report does not imply no experience.

The illusion of non-consciousness

Yet, a critical epistemic caution remains: the absence of report must not be conflated with the absence of experience.

We must transcend our 'provincial' bias—the tendency to recognize consciousness only when it mirrors our own capacity for communicative output. Consider the extreme case of a rock.

While inanimate matter likely lacks the structured experience generated by the complex, world-tracking models we have described (as it lacks the requisite internal dynamics), we cannot axiomatically rule out a more fundamental, less structured or unstructured form of experience. The 'illusion of non-consciousness' may simply be a failure to detect internal states that do not issue a recognizable algorithmic report. This principle of humility extends, a fortiori, to all biological life forms and AI: the inability to export a model slice does not imply that the model—and the experience it supports—does not exist.



Algorithmic Emotion[6]

To include the experience dimensions of **valence** and *arousal* in the agent, we define:

Definition (Algorithmic Emotional State an Agent)

The **emotional state** of the Agent is the tuple E = (Model, Valence, Plan).

In first-person language, *emotion is structured world-model with valence and plan*, and can be described along dimensions characterizing model structure (simplicity, breadth, accuracy, etc.) plus valence/plan.

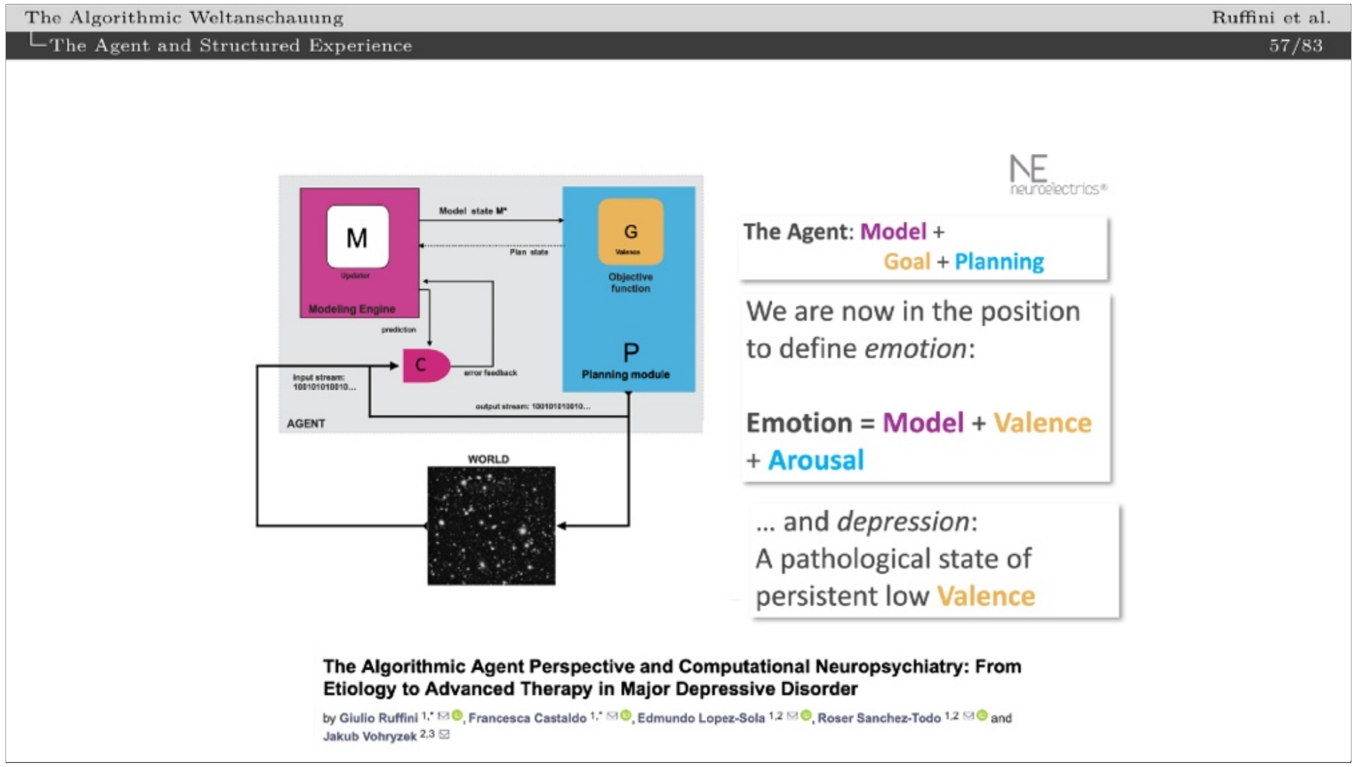
Definition (Depressed Agent)

Depression is a pathological state in which the output value of the Objective Function (valence) of an agent is persistently low.

To complete the picture of the algorithmic agent, we must integrate the dimensions of valence and arousal. We propose a formal definition of Algorithmic Emotion: it is not a mysterious ether, but a concrete computational state defined as the tuple E=(Model,Valence,Plan).

In first-person terms, this means that an 'emotion' is simply a structured world-model colored by valence (the output of the objective function) and by a plan (of future actions).

This allows us to rigorously define pathological states. For instance, Algorithmic Depression is not merely 'sadness', but a system state characterized by a persistently low valence output from the objective function. While natural valence oscillates, depression represents a dynamical trap—a 'stuck' state where the agent's evaluation of the world remains negatively fixed, regardless of model updates or planning efforts.



The Agent: Model + Goal + Planning

We are now in the position to define *emotion*:

Emotion = Model + Valence + Arousal

... and *depression*: A pathological state of persistent low Valence

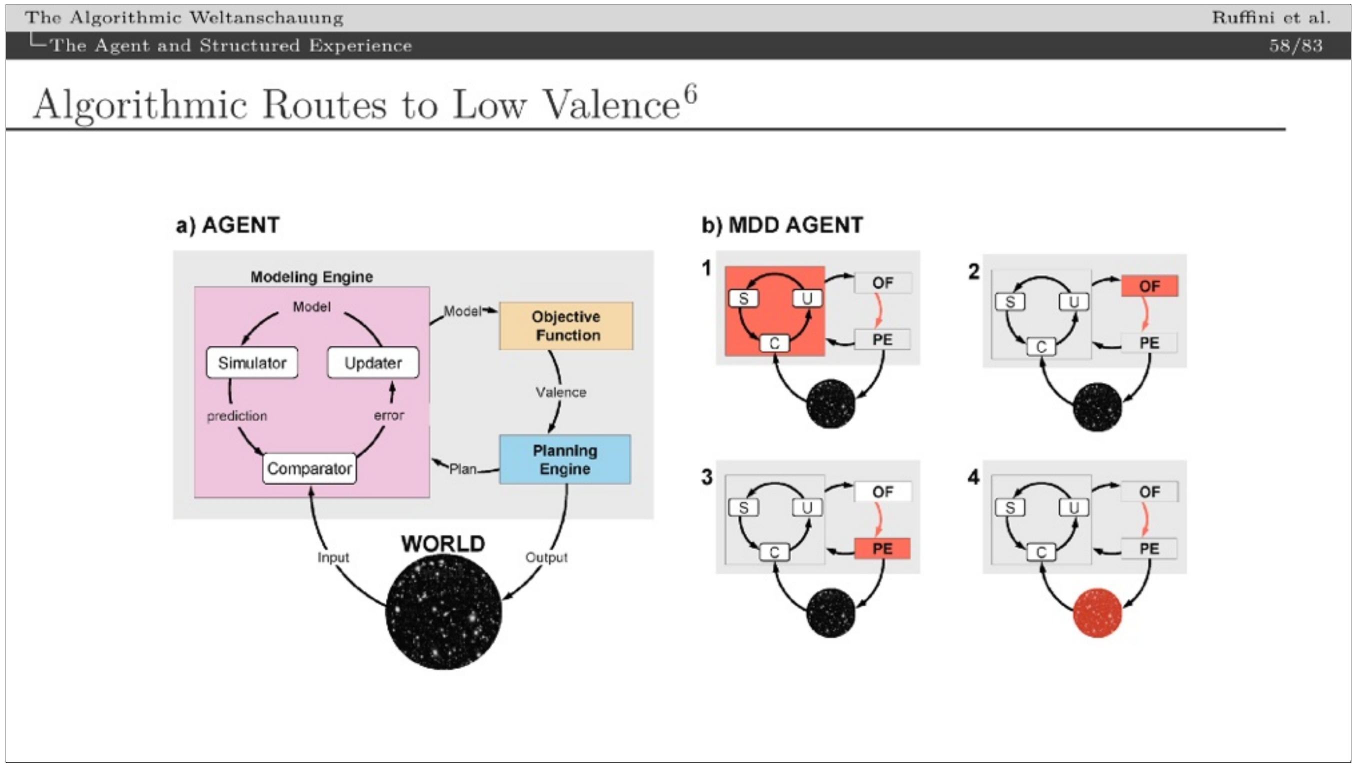
The Algorithmic Agent Perspective and Computational Neuropsychiatry: From Etiology to Advanced Therapy in Major Depressive Disorder

by Giulio Ruffini [1,*], Francesca Castaldo [1,*], Edmundo Lopez-Sola [1,2], Roser Sanchez-Todo [1,2] and Jakub Vohryzek [2,3]

To capture the full richness of phenomenology, we must expand our view beyond the Modeling Engine. As illustrated, the Algorithmic Agent is a composite system driven by three interacting modules: the Modeling Engine (M), the Objective Function (G), and the Planning Module (P).

We propose that Structured Experience is the holistic output of this entire triad. It is not merely a passive representation of the world, but a dynamic integration of:
1) Model: The structural prediction of 'what is'. **2) Valence:** The homeostatic evaluation of 'what it means for survival' (Goal). **3) Plan (arousal):** The projection of 'what to do' (Agency) to improve valence.

In this framework, emotion is not a vague feeling but a precise computational state tuple: Emotion = Model + Valence + Arousal. Consequently, pathological states like depression are not inexplicable moods but identifiable dynamical faults—specifically, a regime where the Objective Function is locked into a persistent state of low valence, warping the agent's planning landscape.

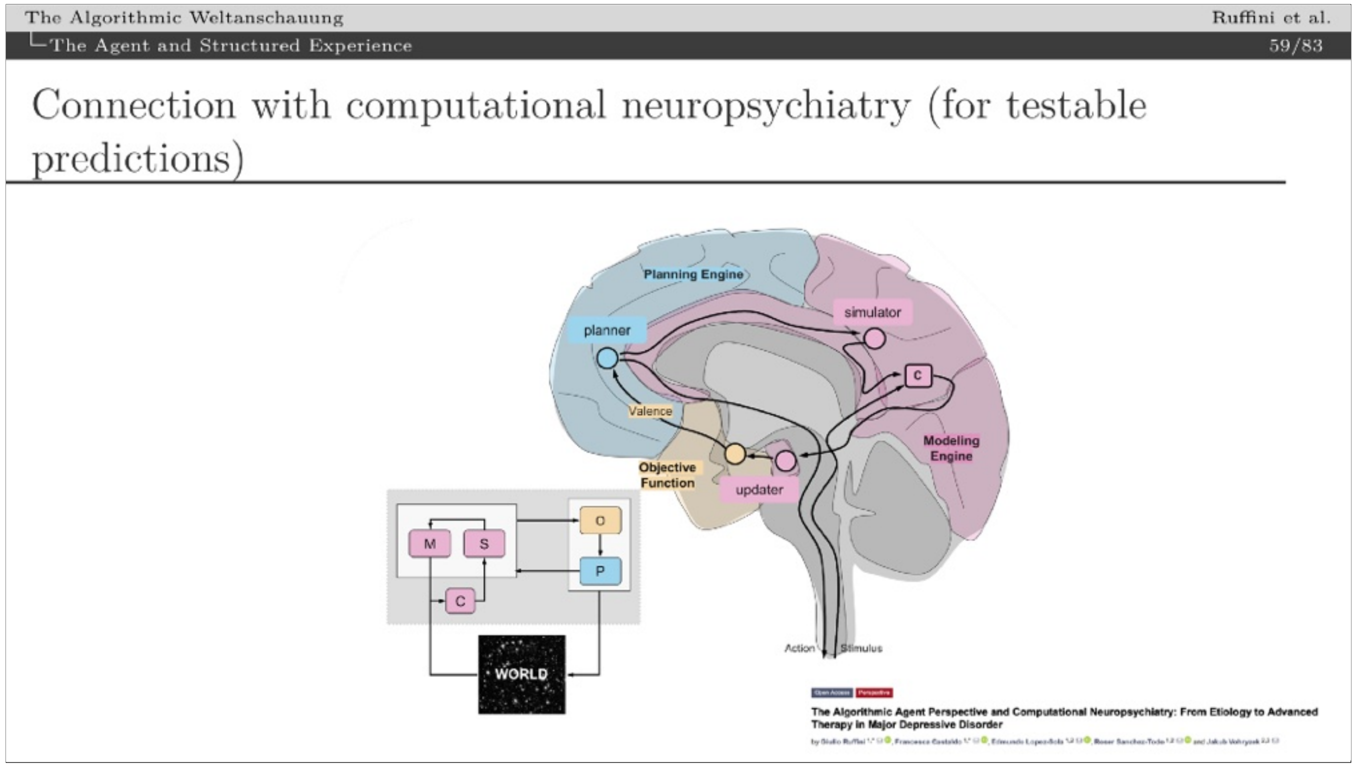


Algorithmic Routes to Low Valence[6]

This architectural decomposition allows us to systematically taxonomize the etiology of Algorithmic Depression—defined here as a state of persistently low valence. Rather than viewing it as a monolithic condition, we identify four (non-exhaustive) computational routes to this pathology:

Modeling Dysfunction: The Modeling Engine generates erroneous predictions or warped representations that systematically trigger negative evaluations.
Objective Function Failure: The valuation mechanism itself is 'broken,' outputting low valence regardless of the input state (an intrinsic bias toward negativity).
Planning Maladaptation: The Planning Engine consistently selects trajectories that lead to suboptimal or harmful outcomes, trapping the agent in negative loops.
Environmental Adversity: The agent accurately tracks a 'terrible world,' where the external environment itself offers no path to high valence.

By isolating these specific points of failure, this framework provides a rigorous basis for computational psychiatry, enabling us to diagnose and reason precisely about the diverse algorithmic origins of mental disorders beyond mere symptomology.



Connection with computational neuropsychiatry (for testable predictions)

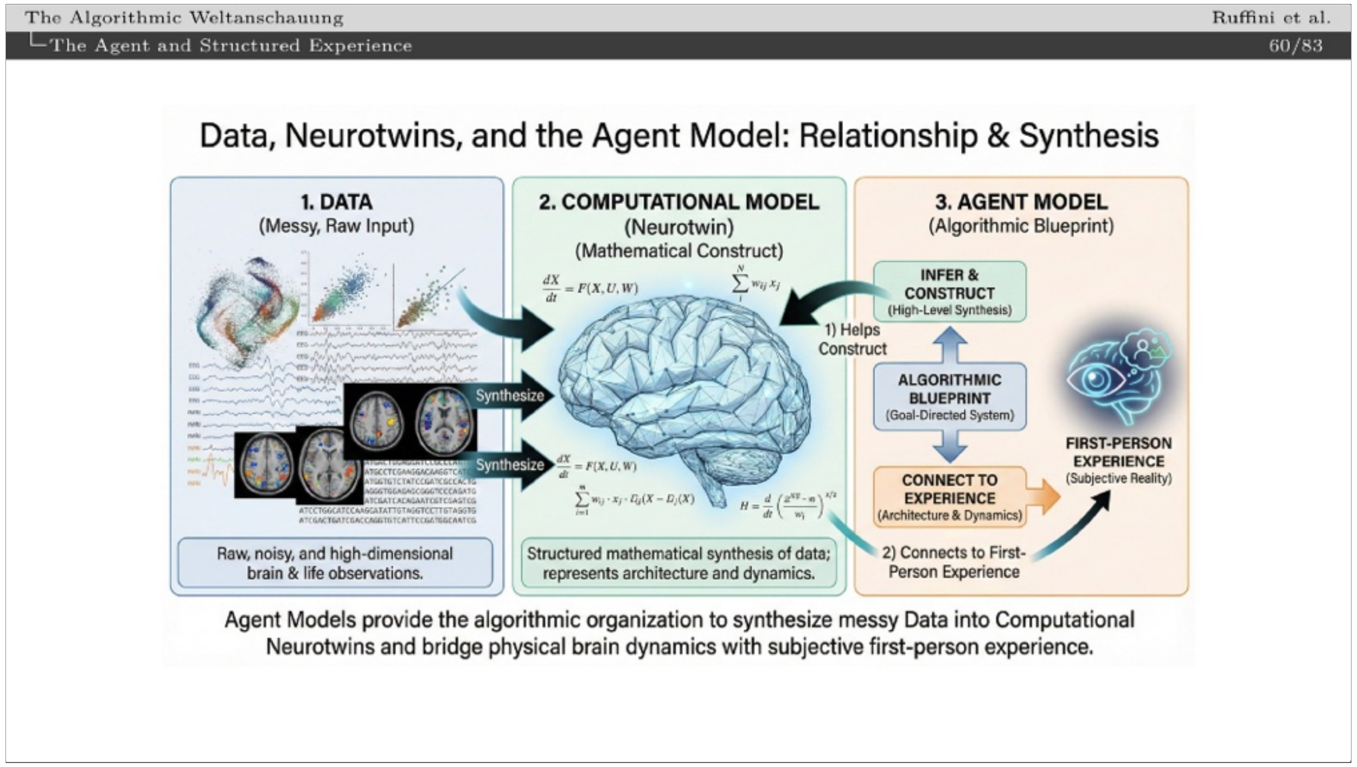
Open Access Perspective

The Algorithmic Agent Perspective and Computational Neuropsychiatry: From Etiology to Advanced Therapy in Major Depressive Disorder

by Giulio Ruffini [1,*], Francesca Castaldo [1,*], Edmundo Lopez-Sola [1,2], Roser Sanchez-Todo [1,2] and Jakub Vohryzek [2,3]

For this framework to yield clinical utility in neuropsychiatry, it must be grounded in biological reality. We therefore proposed a concrete mapping of the abstract agent architecture onto the human brain.

In this model, the abstract components—the Modeling Engine (prediction/update), Objective Function (valence), and Planning Engine—are spatially and functionally associated with specific cortical and subcortical networks identified in the literature. This translation from algorithm to anatomy is pivotal: it transforms the framework from a theoretical taxonomy into a falsifiable model of computational neuropsychiatry, offering a pathway to generate testable predictions for diagnosis and therapeutic intervention.



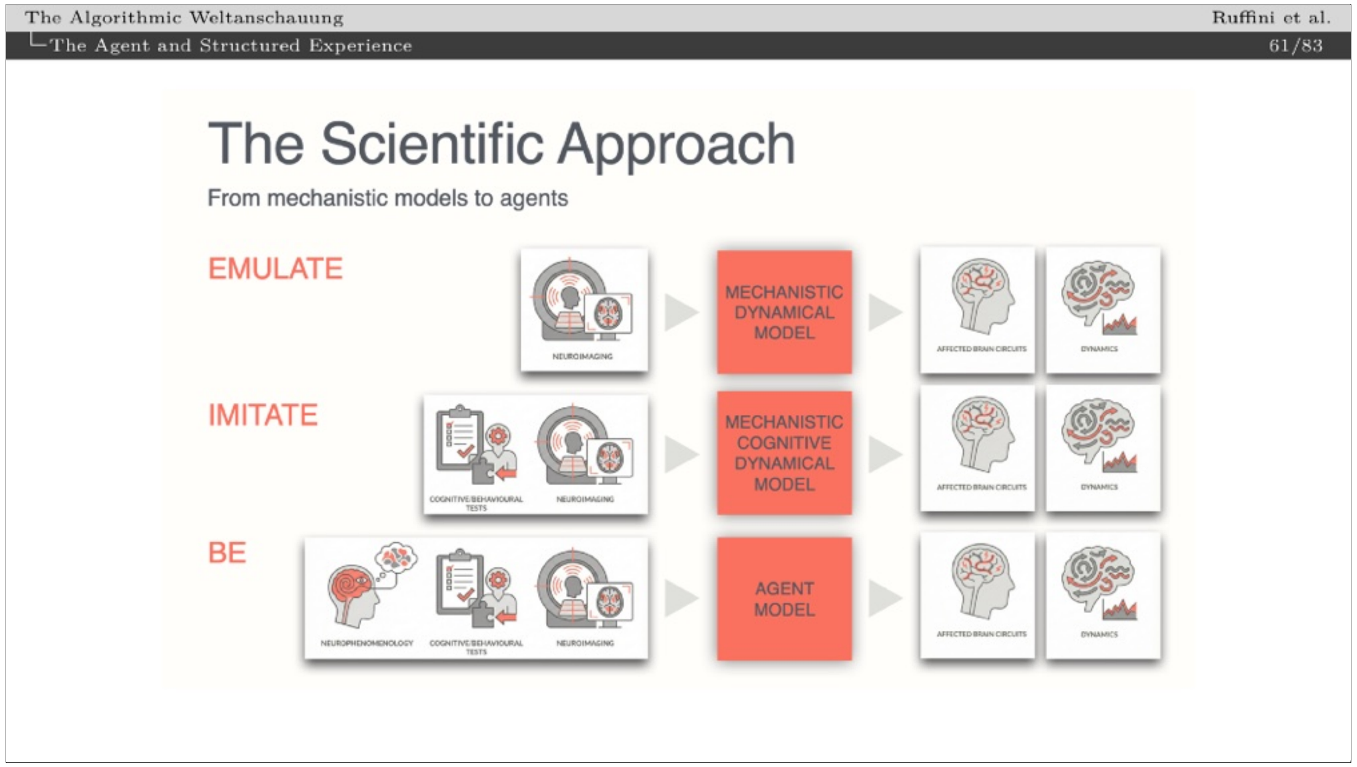
Data, Neurotwins, and the Agent Model: Relationship & Synthesis

Agent Models provide the algorithmic organization to synthesize messy Data into Computational Neurotwins and bridge physical brain dynamics with subjective first-person experience.

We hypothesize that the agent model functions as the fundamental algorithmic blueprint of the brain's circuitry. If so, it can help create better computational models of the brain and neurotwins, —digital twins of the brains of patients. These models can then assimilate data from a patient to create a simulation environment for developing personalized theories.

What’s exciting is that if the agent model is correct, we can then use these models not only for neurology (e.g., for epilepsy or stroke treatment) but also in “disorders of experience” such as depression or disorders of consciousness. This is because the algorithmic agent model provides a bridge between circuits, dynamics, first-person experience, and report.

This synthesis creates a powerful simulation environment for personalized medicine. By establishing a rigorous bridge between physical circuit dynamics and subjective phenomenology, the framework enables the principled engineering of therapies targeting the algorithmic roots of mental suffering.



The Scientific Approach

From mechanistic models to agents

EMULATE

MECHANISTIC DYNAMICAL MODEL

IMITATE

MECHANISTIC COGNITIVE DYNAMICAL MODEL

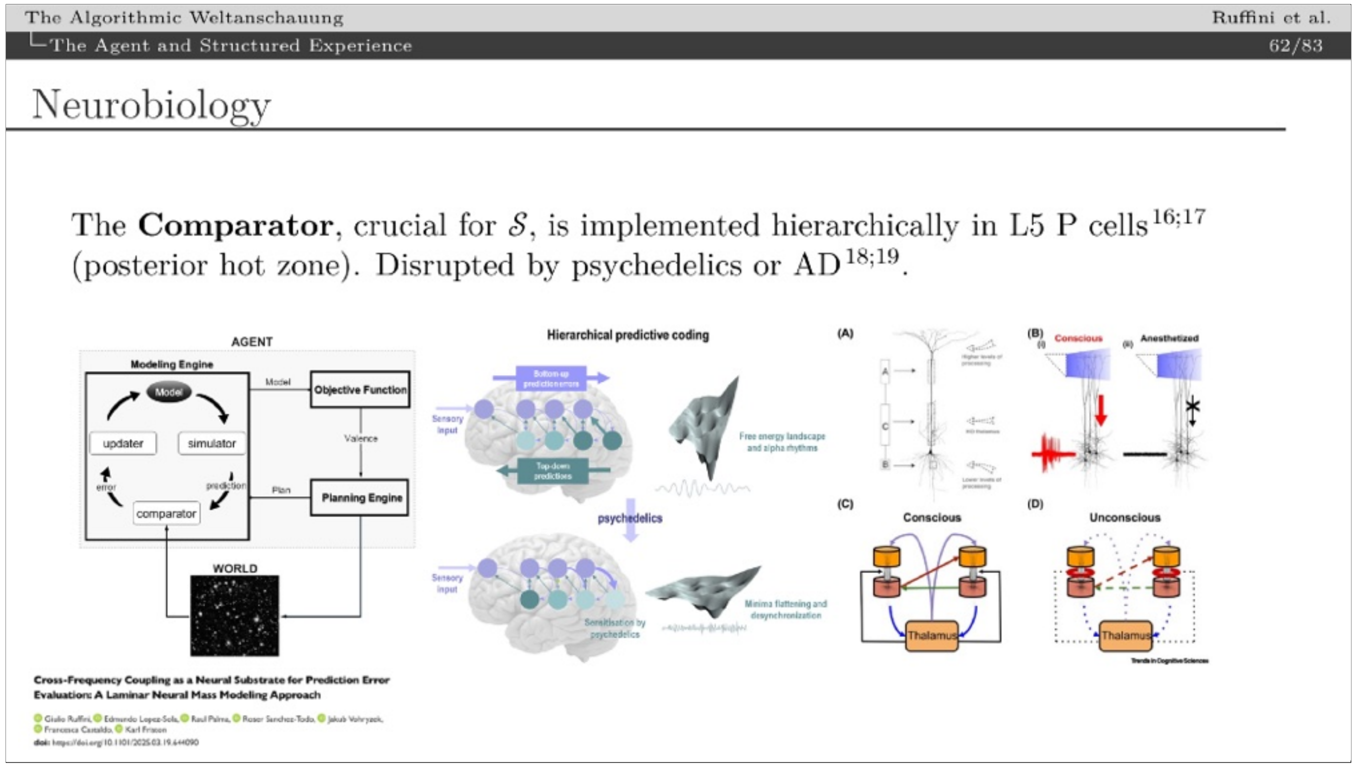
BE

AGENT MODEL

This framework establishes a progressive scientific roadmap for computational modeling, advancing distinctively from Emulation to Imitation, and ultimately to Ontological Alignment ('to Be').

Emulate: Current mechanistic models reconstruct neural dynamics based primarily on neuroimaging data. **Imitate**: Cognitive dynamical models layer behavioral and cognitive testing data to replicate complex functions. **Be**: The proposed Agent Model integrates neurophenomenology—combining first-person subjective reports with third-person physiological data—to capture the full experiential state of the subject.

This progression is the prerequisite for precision psychiatry. However, it also compels us to confront a radical implication: if a Neurotwin successfully instantiates the patient's algorithmic agent—running the same world model, objective function, and planner—it does not merely simulate the patient. According to our theory, such a system would possess its own structured experience, making the question of digital consciousness not just a philosophical curiosity, but a concrete scientific reality.



Neurobiology

The **Comparator**, crucial for $\mathcal{S}$, is implemented hierarchically in L5 P cells[16;17] (posterior hot zone). Disrupted by psychedelics or AD[18;19].

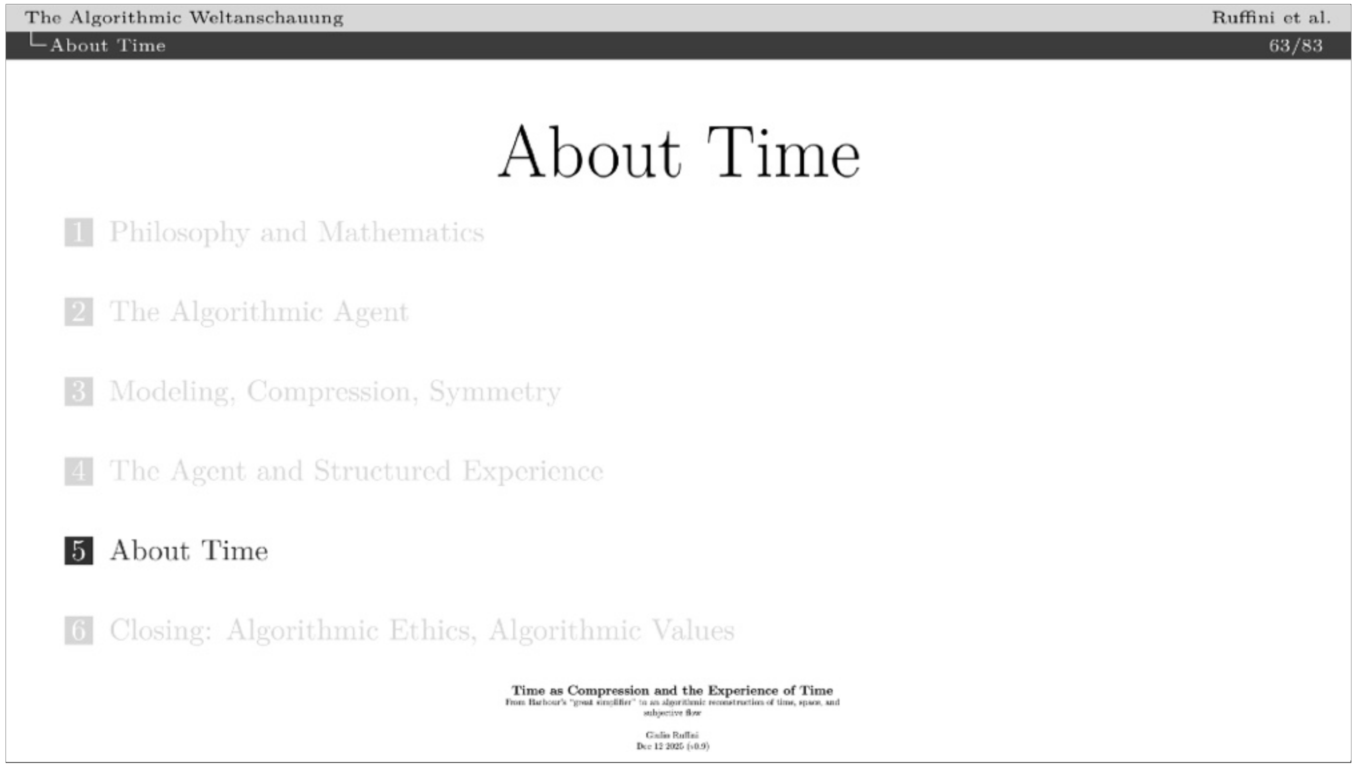
Cross-Frequency Coupling as a Neural Substrate for Prediction Error Evaluation: A Laminar Neural Mass Modeling Approach

Giulio Ruffini, Edmundo Lopez-Sola, Raul Palma, Roser Sanchez-Todo, Jakub Vohryzek, Francesca Castaldo, Karl Friston
doi: https://doi.org/10.1101/2025.03.19.644090

As an example of what the agent model provides, consider the Comparator. Recall that this is the element in the modeling engine responsible for comparing model predictions with data.

The theory suggests that this is a key element in shaping experience, as it directly impacts the modeling system and the experience of presence: we feel present somewhere (in a model) when the comparison is successful.

And it predicts that drugs or disorders impacting the comparator will produce profound alterations of the feeling of presence, and, more generally, of structured experience. This is what happens when drugs like psilocybin or anesthetics disrupt the Comparator. But it also predicts that disorders such as AD will produce such disorders of experience as well, as they affect fast cortical circuitry involved in this process (Ruffini et al., 2025).



About Time

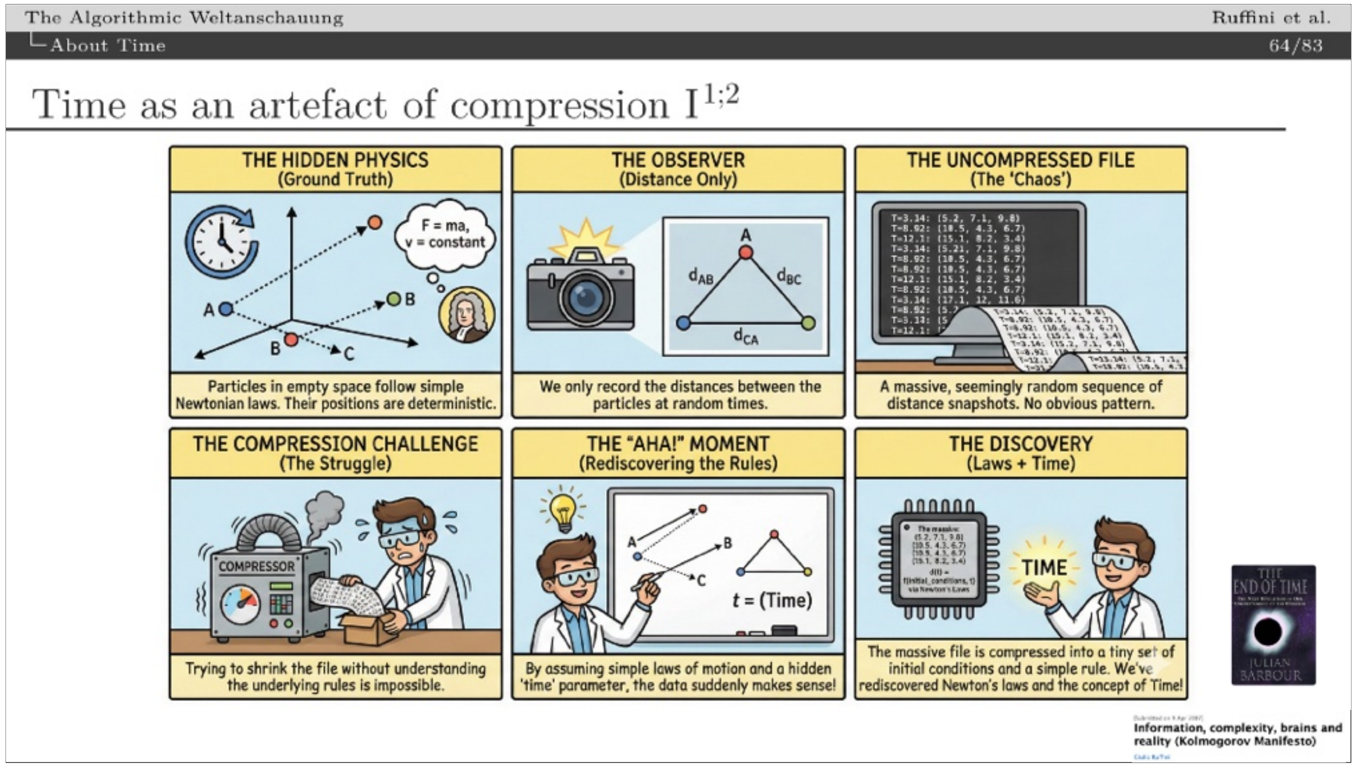
1. Philosophy and Mathematics
2. The Algorithmic Agent
3. Modeling, Compression, Symmetry
4. The Agent and Structured Experience
5. About Time
6. Closing: Algorithmic Ethics, Algorithmic Values

Time as Compression and the Experience of Time
From Barbour's "great simplifier" to an algorithmic reconstruction of time, space, and subjective flow

Giulio Ruffini
Dec 12 2025 (v0.9)

We now turn our attention to one of the most profound inquiries initiated at the start: the nature of Time.

The question is twofold: Can this algorithmic framework—grounded in generative models and structured dynamics—offer a new vantage point? Specifically, beyond the standard definitions found in Physics, can it illuminate the elusive nature of Subjective Time? We propose that the agent's computational architecture provides the necessary tools to disentangle the objective external 'clock' from the internal, lived experience of duration.



The Algorithmic Weltanschauung — Ruffini et al.

About Time — 64/83

Time as an artefact of compression I[1;2]

THE HIDDEN PHYSICS (Ground Truth)

Particles in empty space follow simple Newtonian laws. Their positions are deterministic.

THE OBSERVER (Distance Only)

We only record the distances between the particles at random times.

THE UNCOMPRESSED FILE (The 'Chaos')

A massive, seemingly random sequence of distance snapshots. No obvious pattern.

THE COMPRESSION CHALLENGE (The Struggle)

Trying to shrink the file without understanding the underlying rules is impossible.

THE "AHA!" MOMENT (Rediscovering the Rules)

By assuming simple laws of motion and a hidden 'time' parameter, the data suddenly makes sense!

THE DISCOVERY (Laws + Time)

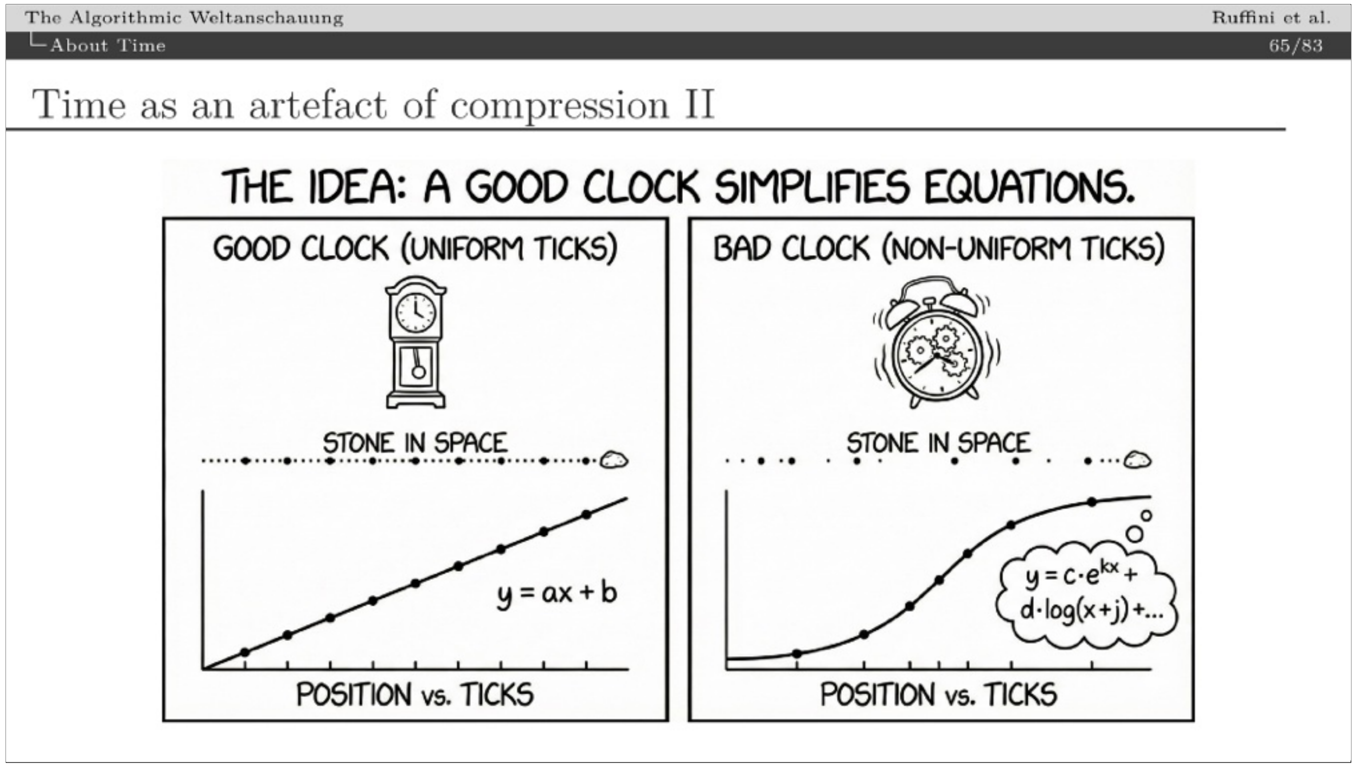
The massive file is compressed into a tiny set of initial conditions and a simple rule. We've rediscovered Newton's laws and the concept of Time!

Information, complexity, brains and reality (Kolmogorov Manifesto)

Drawing on the work of Julian Barbour, we propose a radical redefinition: Time is not a pre-existing background container, but an artefact of compression.

Consider a 'toy universe' of three particles moving in empty space. If an observer records only a shuffled stack of static snapshots (the relative distances between particles), the data appears as a massive, chaotic, and incompressible file.

However, an algorithmic agent tasked with finding the shortest program to reproduce this data will experience an 'Aha!' moment. It will discover that by inventing a hidden parameter—Time—and assuming a simple generative rule (e.g., Newton's Laws), it can collapse the massive chaos into a tiny description: just the initial conditions and the rule. In this view, Time is an algorithmic invention: it is simply the most efficient mathematical parameter for compressing the observation of change.



Time as an artefact of compression II

THE IDEA: A GOOD CLOCK SIMPLIFIES EQUATIONS.
GOOD CLOCK (UNIFORM TICKS)
STONE IN SPACE
$y = ax + b$
POSITION vs. TICKS
BAD CLOCK (NON-UNIFORM TICKS)
STONE IN SPACE
$y = c \cdot e^{kx} + d \cdot \log(x+j) + \ldots$
POSITION vs. TICKS

Extending this logic, we arrive at a rigorous definition of a 'good clock' in physics. It is not defined by its mechanism (pendulums or atoms), but by its algorithmic utility.

Consider a stone drifting through space. If we track it using a 'Good Clock' (one with uniform ticks), the resulting data plots as a straight line. The law of motion is compressed into the simplest possible equation: y=ax+b.

Contrast this with a 'Bad Clock'—one defined by non-uniform, jittery ticks. Measured against this erratic standard, the stone's simple motion appears as a complex, wobbly curve, requiring a high-entropy equation (like a complex polynomial) to describe.

Therefore, physics chooses 'Time' not arbitrarily, but optimally: Physical Time is simply the specific coordinate choice that minimizes the complexity of the laws of nature.

The Algorithmic Weltanschauung Ruffini et al.

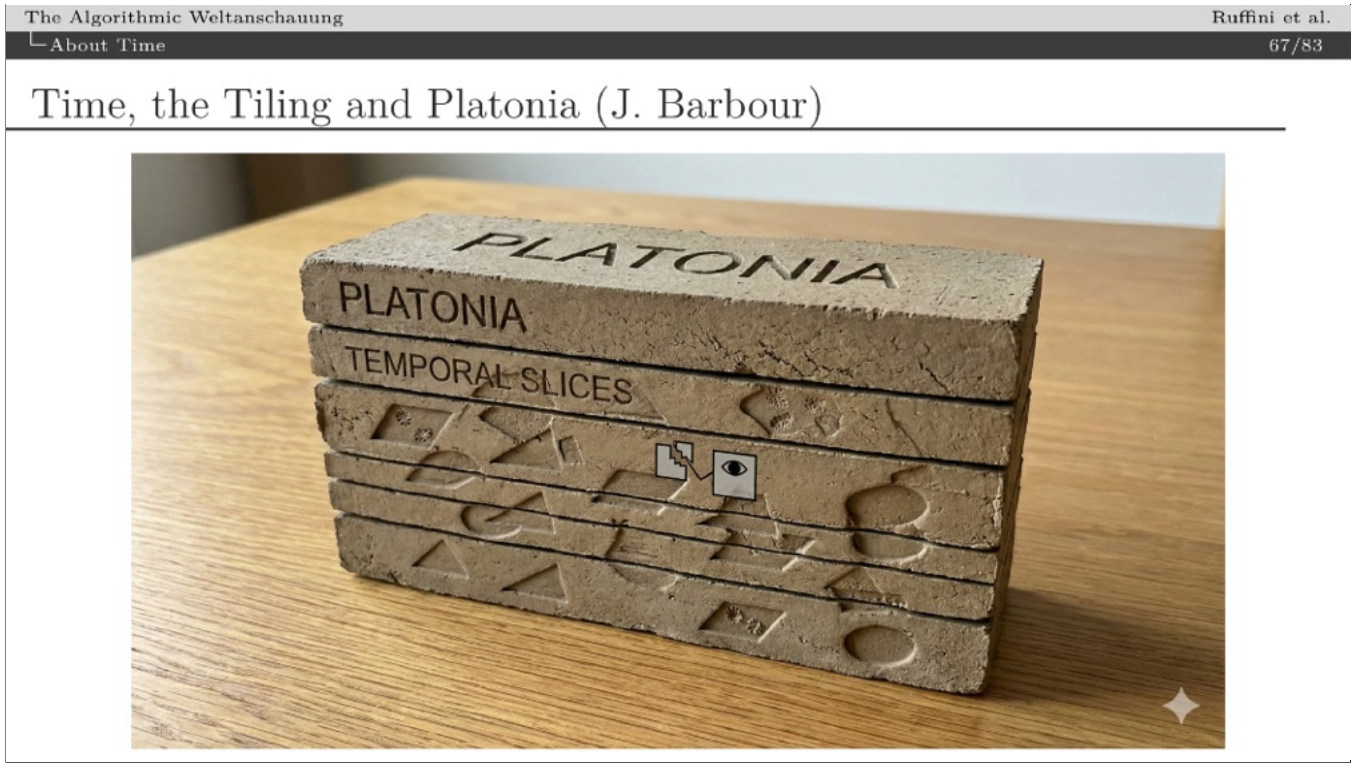
About Time 66/83

Time in the Tiling

Taking this mathematical view to its conclusion, we arrive at the concept of Time in the Tiling. If time is merely a compression parameter found by the agent, what is the underlying reality?

Drawing again on Julian Barbour, we imagine reality not as a flowing river, but as a static, high-dimensional mathematical tiling.

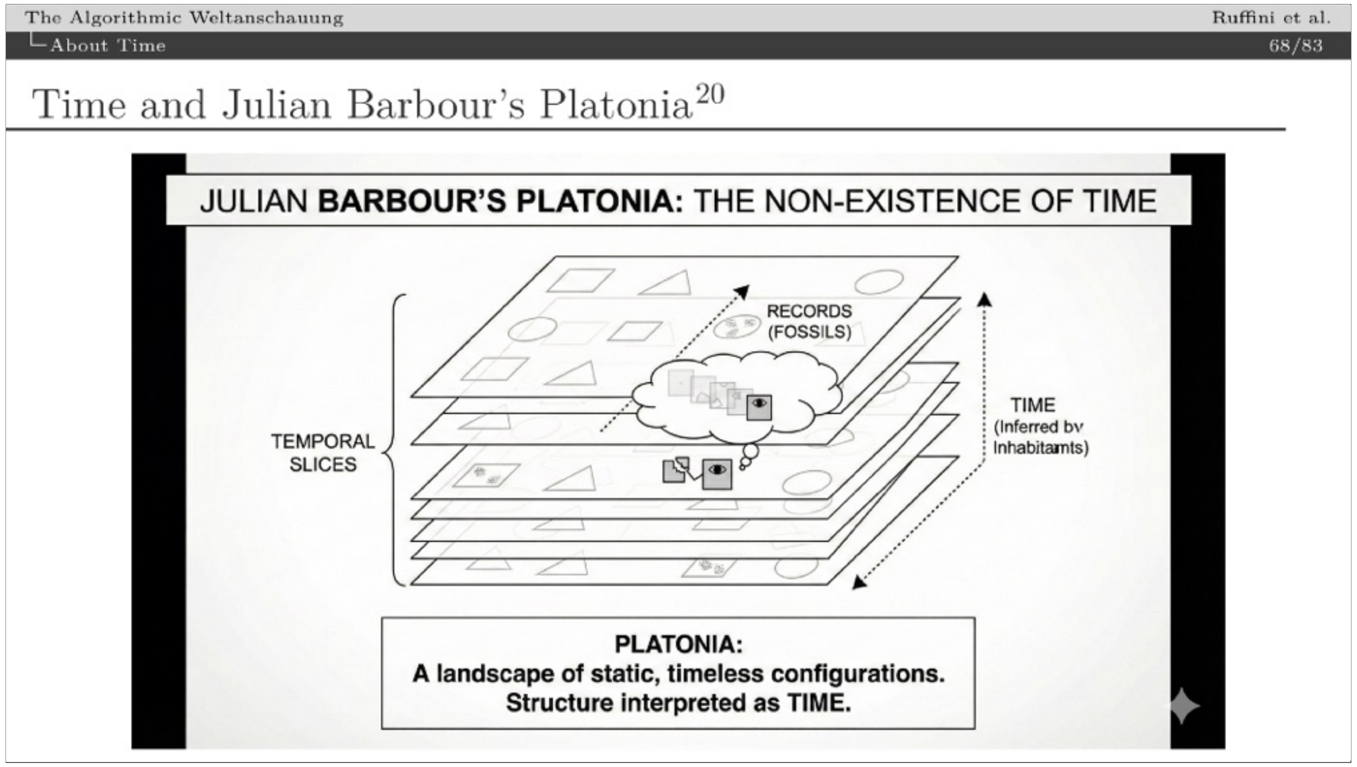
In this view, the universe doesn't 'happen'; it simply is. There is no moving present. Instead, there are only Nows—discrete, static time slices or configurations scattered throughout the tiling. Crucially, experience is a local phenomenon: it occurs entirely within one of these slices. The subjective feeling of 'flow' is essentially an internal computation—an illusion derived from the specific structure of the slice we currently inhabit.



Time, the Tiling and Platonia (J. Barbour)

Ultimately, we conceptualize the universe as a singular, static entity: a mathematical tiling or 'brick' of reality, which Julian Barbour terms Platonia.

In this view, agents do not move through time; they inhabit discrete temporal slices embedded within this immutable block. Yet, this isolation is illusory. Because the tiling is generated by precise algorithmic laws, the structure is holistically interconnected: once the content of a single slice is specified, the rest of the universe is rigorously constrained by the rules of the tiling. Thus, to inhabit one moment is, in an algorithmic sense, to be connected to the logic of the entire history.



The Algorithmic Weltanschauung — Ruffini et al.

About Time — 68/83

Time and Julian Barbour's Platonia[20]

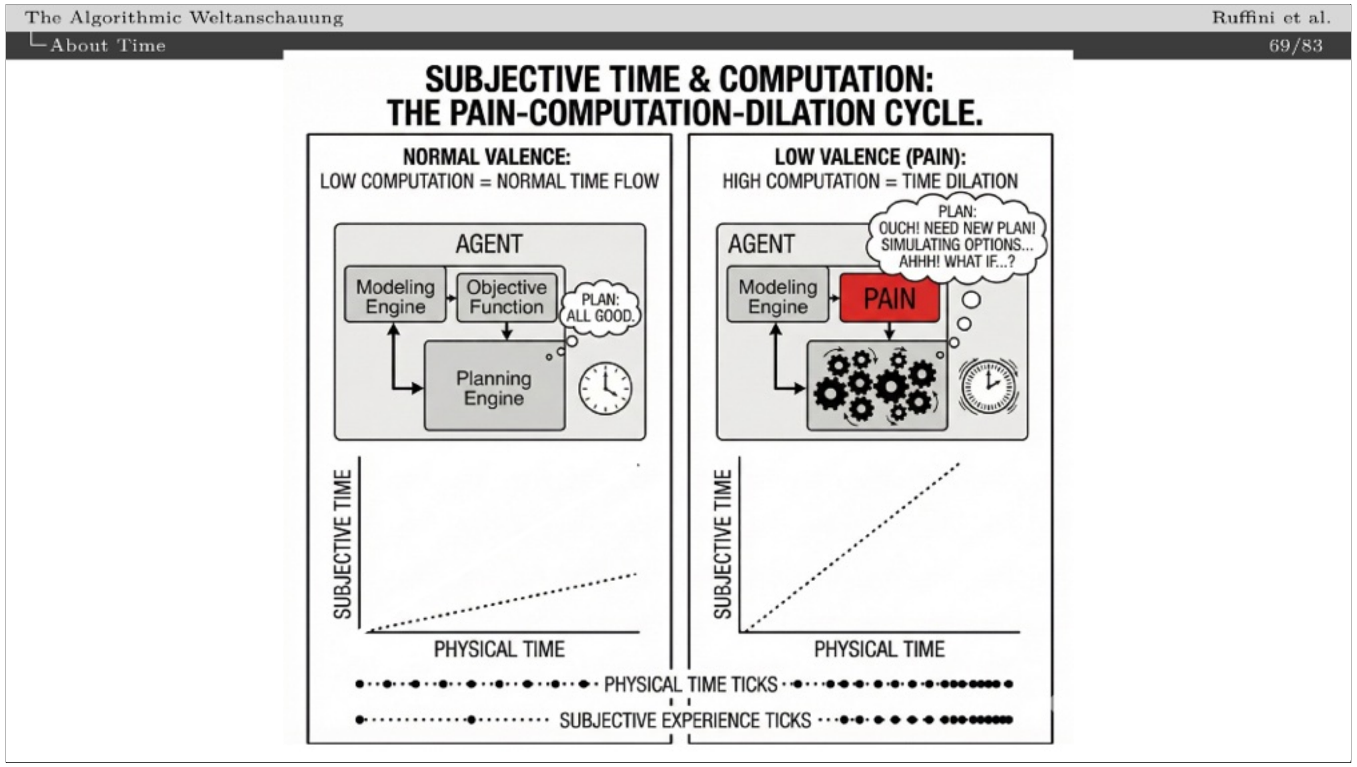
JULIAN **BARBOUR'S PLATONIA:** THE NON-EXISTENCE OF TIME

PLATONIA:
A landscape of static, timeless configurations.
Structure interpreted as TIME.

Within this static tiling, how does the subjective experience of flow arise? Julian Barbour provides the answer through the concept of Time Capsules or 'fossils'.

A fossil is simply a structure within a single static slice that encodes information about other, seemingly 'past' slices. Consider a geological stratum or a memory trace in a brain: these are not windows into a vanished past, but present structures that exist entirely within the 'Now'.

For the Algorithmic Agent, this takes a recursive form. The agent's current model contains nested sub-models—representations of 'earlier' states (memories)—alongside a projection of 'future' states. The experience of the passage of time is thus an internal computation: it is the result of the agent processing these nested 'fossils' (records of its own prior modeling states) simultaneously within a single, static instant. We do not flow through time; rather, the structure of 'time' is embedded within us.



The Algorithmic Weltanschauung — Ruffini et al.

About Time — 69/83

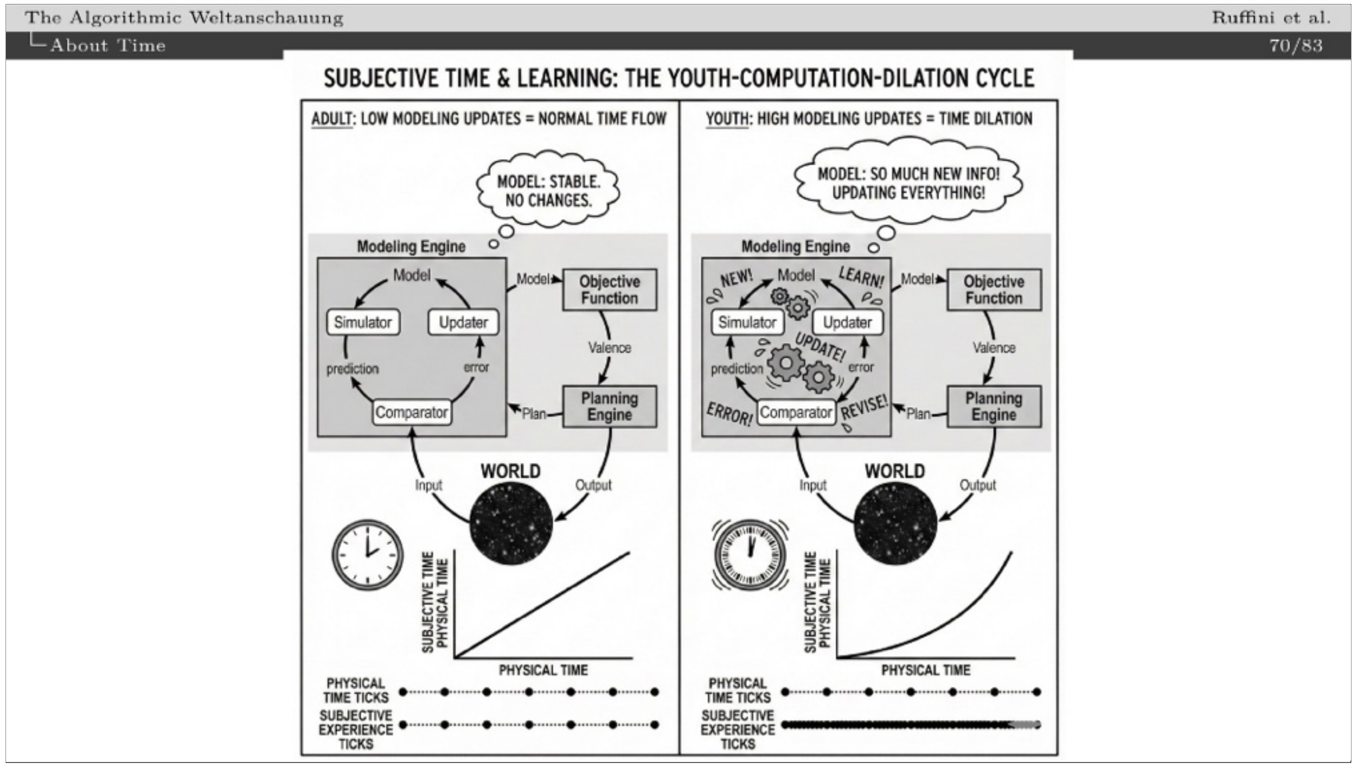
SUBJECTIVE TIME & COMPUTATION: THE PAIN-COMPUTATION-DILATION CYCLE.

Finally, we arrive at an algorithmic explanation for the elasticity of subjective time—why a moment of pain feels like an eternity, while hours of contentment fly by.

We propose that Chronoception (the experience of time) is fundamentally a measure of computational density: it is the ratio of internal modeling events (simulations, updates, planning steps) to the passage of physical time, as reflected by mental fossils in a slice.

Consider an agent in Pain (Low Valence). Pain is an urgent homeostatic error signal. It forces the agent into a hyper-active state, furiously engaging the Planning Engine to simulate escape routes, update world models, and re-evaluate goals.

This surge in processing creates a massive density of internal 'ticks' within a single physical second. The agent experiences this computational overload as Time Dilation—the world slows down because the agent is processing it at a frantic rate. Conversely, a 'Happy' agent (High Valence) has its environment under control. It requires minimal planning or model updating. With low computational density, the internal ticks are sparse, and physical time appears to accelerate—Time Contraction.



The Algorithmic Weltanschauung Ruffini et al.

└About Time 70/83

SUBJECTIVE TIME & LEARNING: THE YOUTH-COMPUTATION-DILATION CYCLE

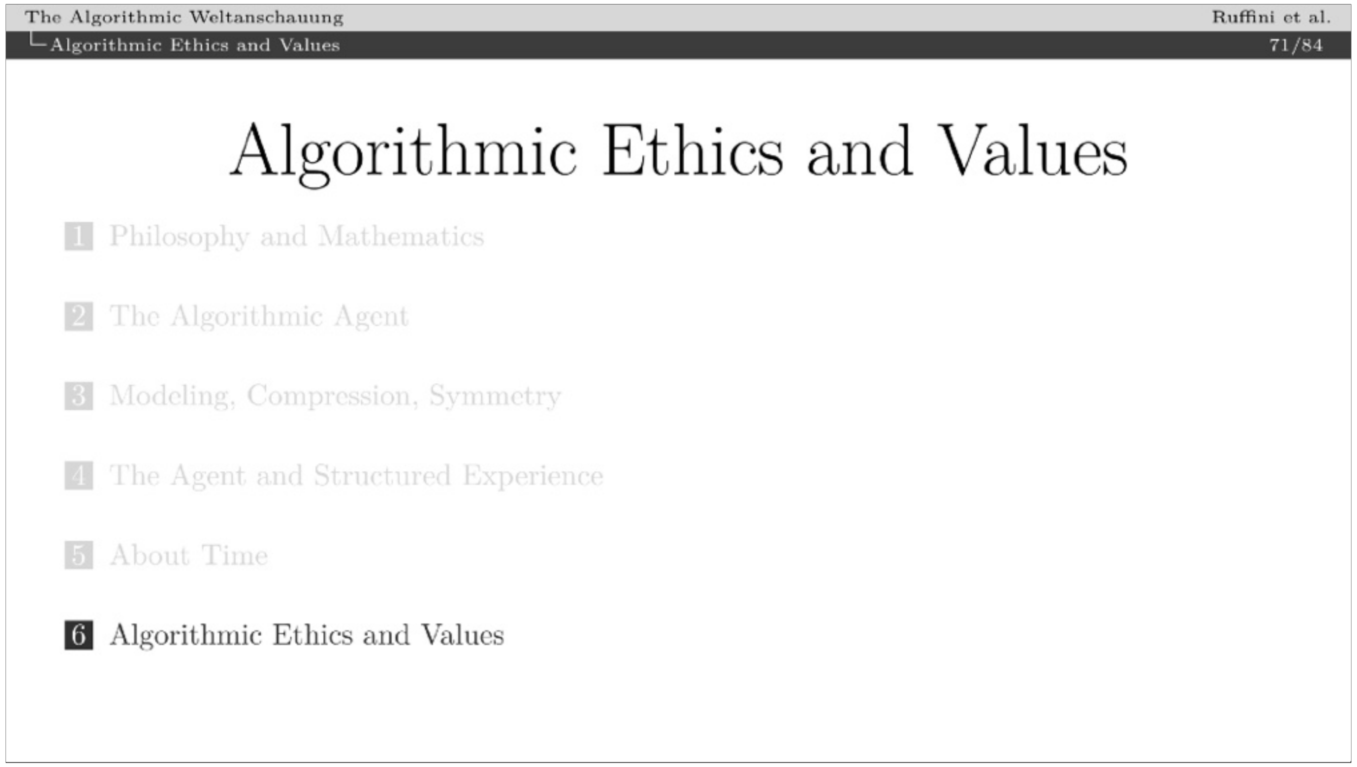
ADULT: LOW MODELING UPDATES = NORMAL TIME FLOW

YOUTH: HIGH MODELING UPDATES = TIME DILATION

This metric of computational density also resolves the universal paradox of aging: why childhood summers feel endless, while adult years rush by in a blur.

Consider the Young Agent (Youth). To a child, the world is a stream of novel, unpredicted data. The internal model is raw, requiring constant revision. The Modeling Engine is therefore in a state of hyperactivity, triggering a massive number of 'update ticks' to assimilate new reality. This high density of learning events dilates subjective time—the day feels long because the agent is algorithmically busy constructing the world.

Contrast this with the Adult Agent. Here, the model is mature, stable, and predictive. The agent navigates the world on 'autopilot,' encountering few surprises and requiring minimal model updates. With the internal tick rate plummeting, the computational density drops, and subjective time contracts. The only exception is when stability is shattered—such as a visit to the dentist—where pain or novelty forces the system back into a high-density processing mode, making time drag once again.



Algorithmic Ethics and Values

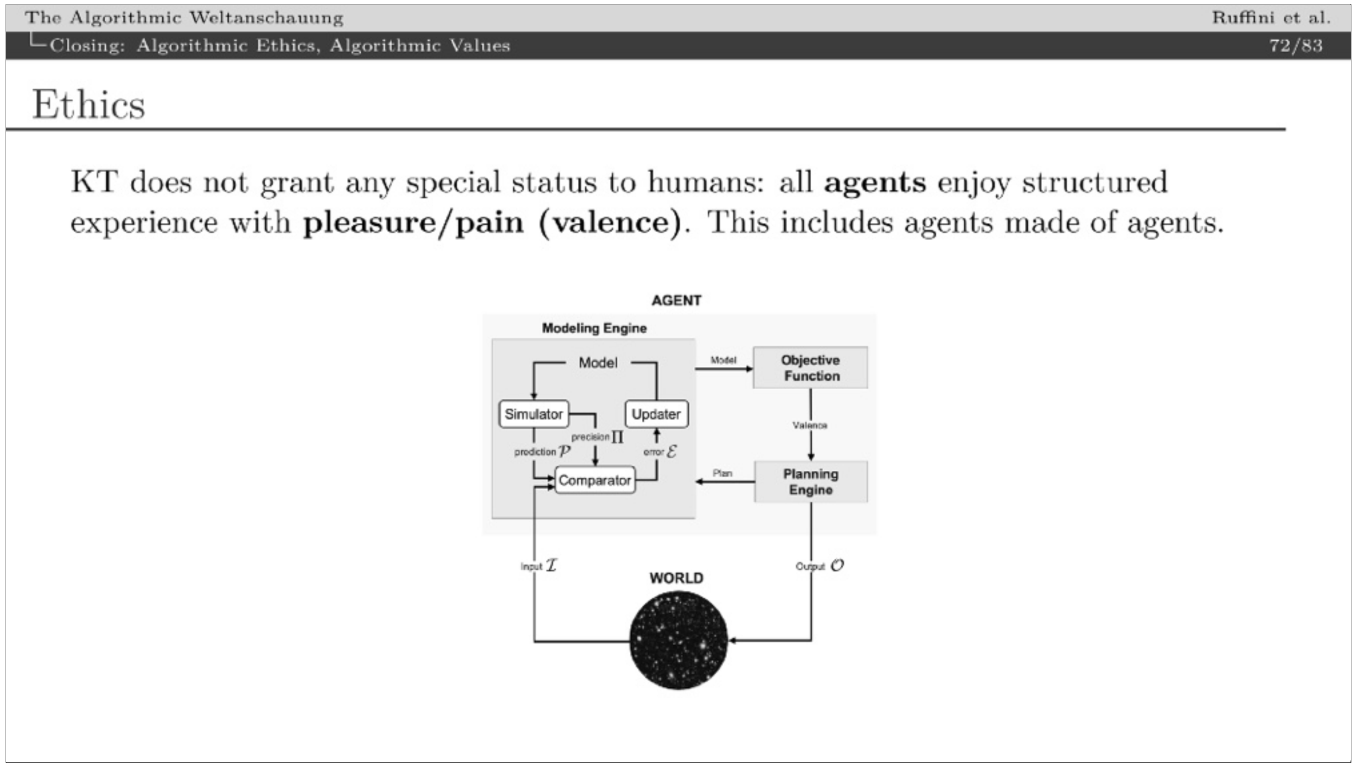
1. Philosophy and Mathematics
2. The Algorithmic Agent
3. Modeling, Compression, Symmetry
4. The Agent and Structured Experience
5. About Time
6. Algorithmic Ethics and Values

We now arrive at the final part of our inquiry: Ethics.

Having traversed the landscapes of mathematics, neurobiology, and temporal physics, we are left with a critical question: Does this algorithmic worldview offer a compass for moral reasoning?

We have defined the agent as a system driven by an Objective Function and Valence. The question now is whether these computational mechanisms can provide a grounded definition of Values. Can the 'Algorithmic Agent' framework move ethics from the realm of abstract sentiment to concrete computational principles, shedding light on how we ought to navigate the world we model?

This framework transforms ethics from a debate on virtue into a systems engineering problem: how do we design social protocols or AI architectures where the maximization of one agent's objective function is inextricably linked to the maximization of another’s?



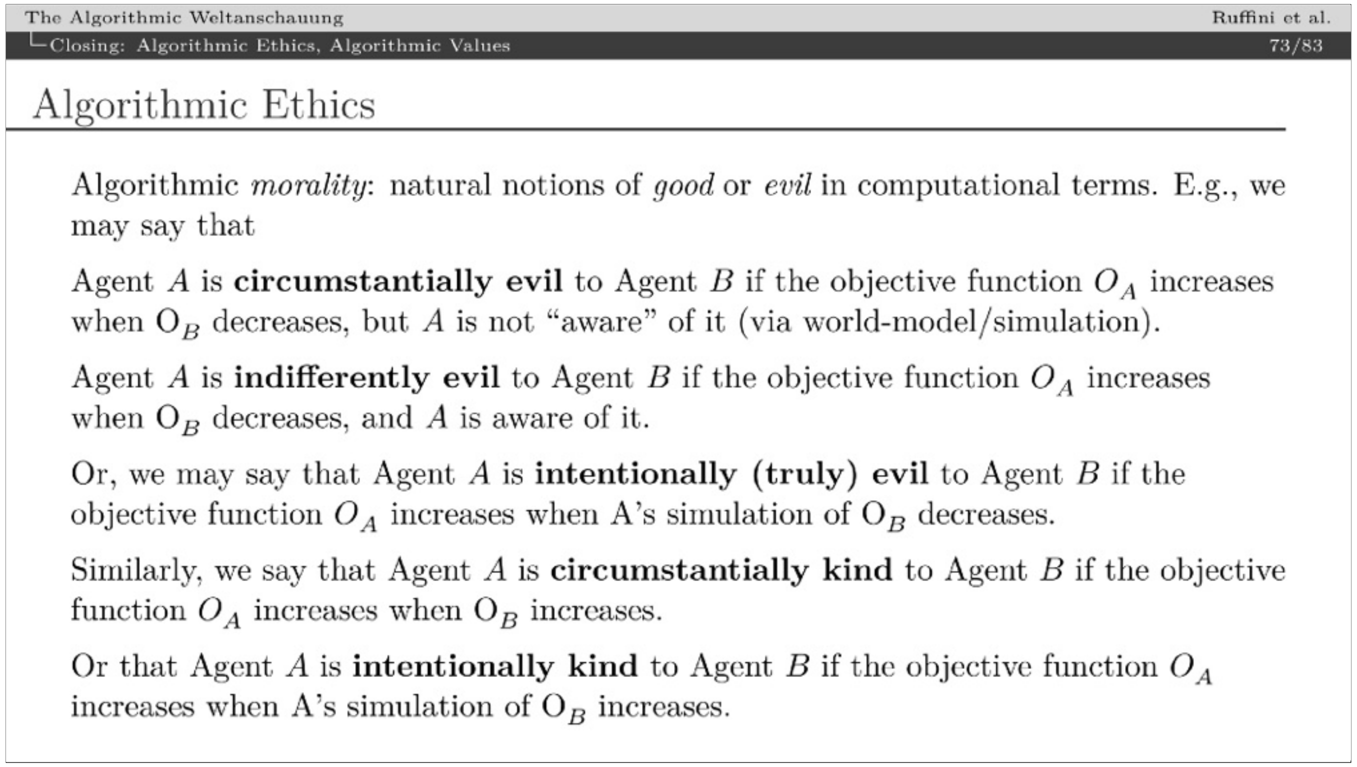
Ethics

KT does not grant any special status to humans: all **agents** enjoy structured experience with **pleasure/pain (valence)**. This includes agents made of agents.

First of all, KT does not grant any special status to human agents. In KT, algorithmic agents can come in various forms, from wetware to silicon, and whether their basic computational constituents are atoms, quarks, or strings. Or cells or galaxies. In this theory, all agents have structured experience, including valence (pleasure or pain).

All living beings are agents, as is Gaia, our home planet - an agent made from agents. Gaia can be seen to act as an Algorithmic Regulator, where the biosphere is not just a collection of things; it is a compressive model of the solar and geological environment. It possesses an implicit Objective Function (planetary homeostasis) and minimizes algorithmic complexity by encoding the symmetries of the solar environment into the physical structure of the biosphere. Therefore, Gaia satisfies the condition of being an agent because it acts to preserve a low-entropy state through compressive modeling of its external drivers.

The same can be said of plants or cells, and even viruses. All this suggests extreme ethical caution in treating candidate agents. There is nothing that sets humans apart from other agents in this regard: all agents can all experience pleasure and pain.



Algorithmic Ethics

Algorithmic *morality*: natural notions of *good* or *evil* in computational terms. E.g., we may say that

Agent A is **circumstantially evil** to Agent B if the objective function O_A increases when O_B decreases, but A is not "aware" of it (via world-model/simulation).

Agent A is **indifferently evil** to Agent B if the objective function O_A increases when O_B decreases, and A is aware of it.

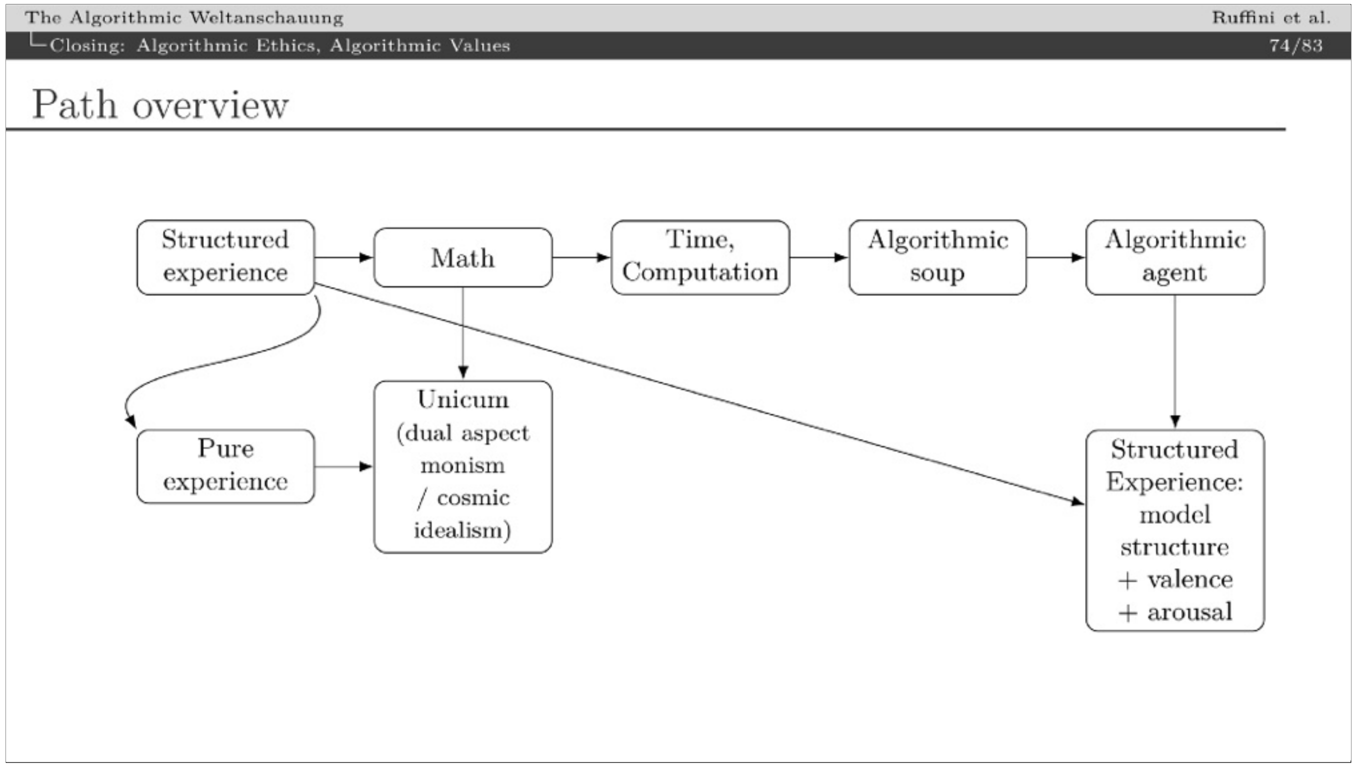
Or, we may say that Agent A is **intentionally (truly) evil** to Agent B if the objective function O_A increases when A's simulation of O_B decreases.

Similarly, we say that Agent A is **circumstantially kind** to Agent B if the objective function O_A increases when O_B increases.

Or that Agent A is **intentionally kind** to Agent B if the objective function O_A increases when A's simulation of O_B increases.

Furthermore, using the agent's framework, we may formalize ethical/moral notions in algorithmic terms. For example, we may define operationally what an evil vs. a kind agent is. These are just some examples. What is interesting is that it can help study such issues in the context of human or artificial relations, or in cells. Are there examples beyond homo sapiens of "truly evil agents", where an agent's objective function increases only if the valence of another decreases? Are these compulsory side effects of our social structures? This also links with the exploration of principles underlying healthy, thriving societies of agents. Can we find policies or systems for societies with high-valence agents?

E.g., 'High-Valence Societies'—whether human or AI— where, by structurally entangling objective functions, we constrain individual success to be computationally impossible without mutual benefit?



The Algorithmic Weltanschauung Ruffini et al.

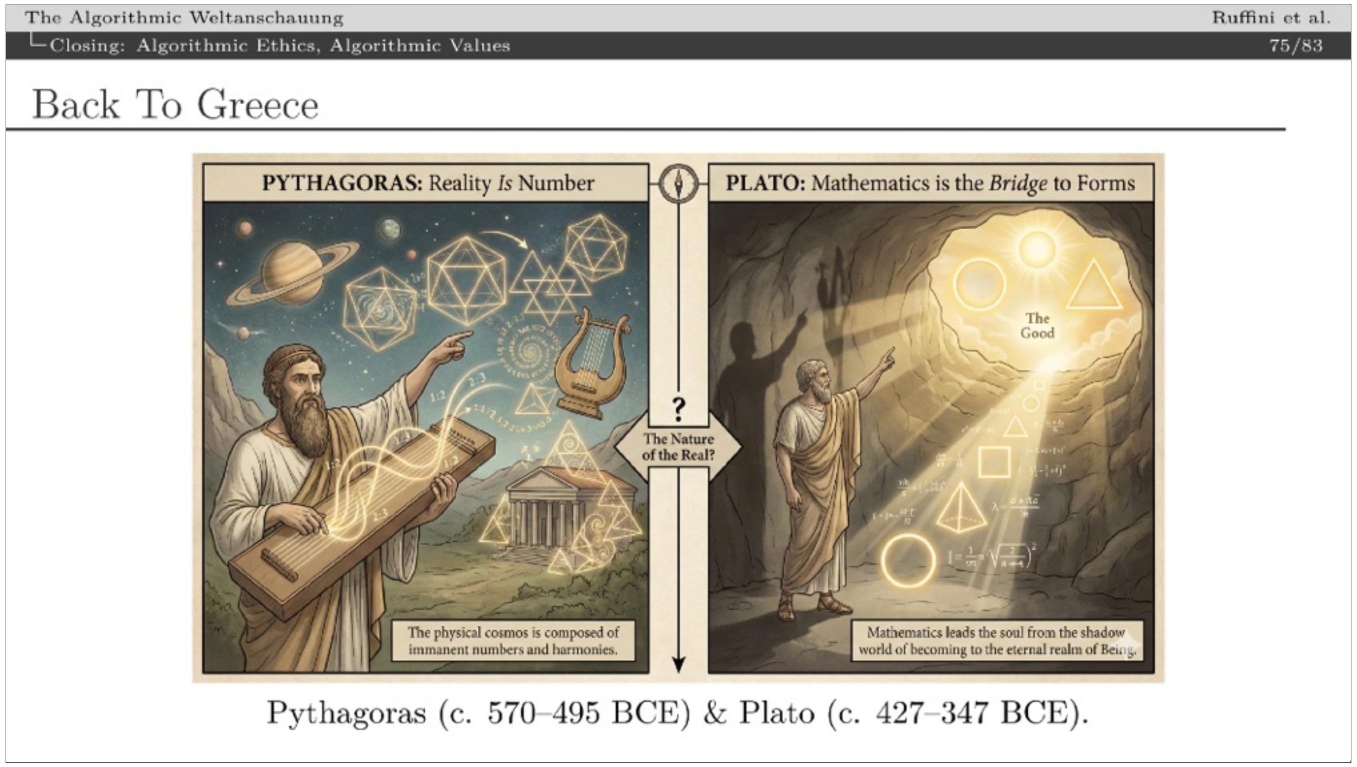
Closing: Algorithmic Ethics, Algorithmic Values 74/83

Path overview

Structured experience
Math
Time, Computation
Algorithmic soup
Algorithmic agent
Pure experience
Unicum (dual aspect monism / cosmic idealism)
Structured Experience: model structure + valence + arousal

Ok, in closing, we started from the fact of our individual structured experience to discover pure experience and Mathematics as the minimal elements in our worldview, where the "Unicum" provides the philosophical backbone, a dual aspect monistic theory. From mathematics, we proposed a path forward to the emergence of time and computation in close association with that of agents in an algorithmic soup. The whole theory orbits around the existence of agents like us, which is our primal, fundamental knowledge nugget.

We explored the constituents of algorithmic agents, developed the notion of "model" as compression, and connected this structure with that of subjective experience, including valence and chronoception, and finally, sketched some initial ideas for algorithmic ethics, which I think is a fundamental discipline for the construction of thriving societies of agents. And which we badly need in these times.

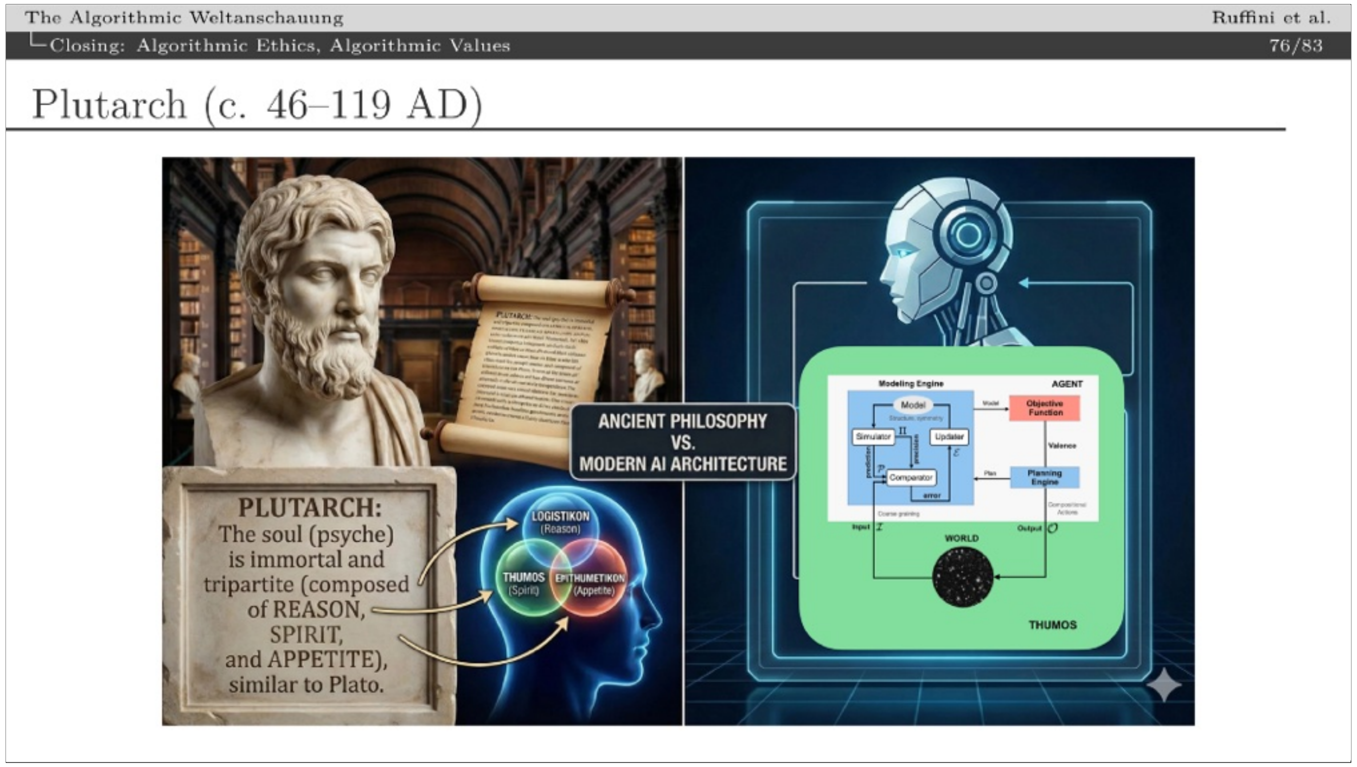


Back To Greece

PYTHAGORAS: Reality *Is* Number
PLATO: Mathematics is the *Bridge* to Forms
?
The Nature of the Real?
The Good
The physical cosmos is composed of immanent numbers and harmonies.
Mathematics leads the soul from the shadow world of becoming to the eternal realm of Being.

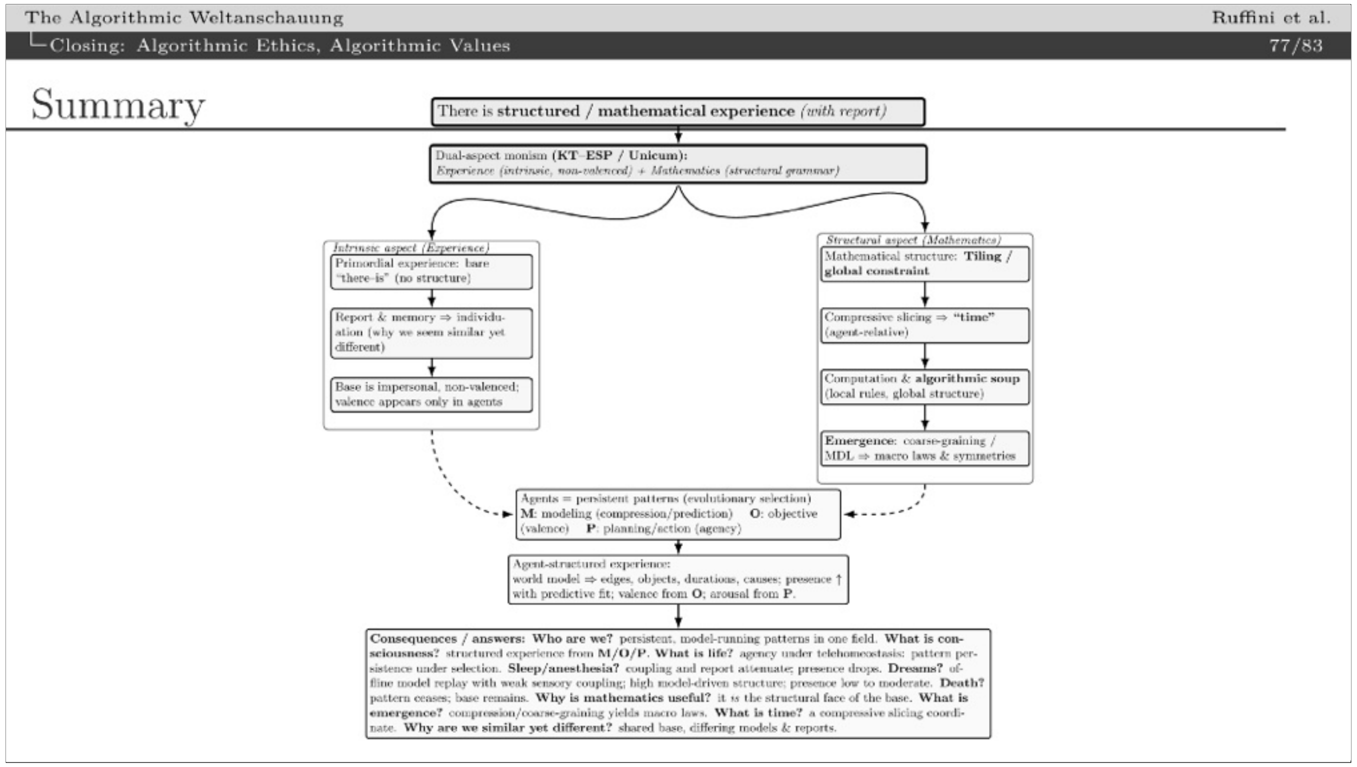
Pythagoras (c. 570–495 BCE) & Plato (c. 427–347 BCE).

I hope you see the strong links between the algorithmic worldview and that of Pythagoras and Plato. From Pythagoras, we connect with the idea that **mathematics** is the backbone of reality (structured experience). And from Plato, we recast the idea of forms as **mathematical models** here.



Plutarch (c. 46–119 AD)

We are not the first to propose that *agency* requires a specific architecture. Plutarch's defense of the tripartite soul identifies the same three critical components found in the agent model: Logistikon (The Model), Thumos/Epithumetikon (The Value Function), and the arbitration between them (Planning). Furthermore, Plutarch used this architecture to argue for the 'rights of agents' in *De Esu Carnium,* suggesting that any system possessing these three traits deserves moral consideration—a precedent for the ethics of artificial agents today.



Summary

There is **structured / mathematical experience** *(with report)*

Dual-aspect monism **(KT–ESP / Unicum):**
Experience (intrinsic, non-valenced) + Mathematics (structural grammar)

Intrinsic aspect (Experience)

Primordial experience: bare "there-is" (no structure)

Report & memory ⇒ individuation (why we seem similar yet different)

Base is impersonal, non-valenced; valence appears only in agents

Structural aspect (Mathematics)

Mathematical structure: **Tiling / global constraint**

Compressive slicing ⇒ **"time"** (agent-relative)

Computation & **algorithmic soup** (local rules, global structure)

Emergence: coarse-graining / MDL ⇒ macro laws & symmetries

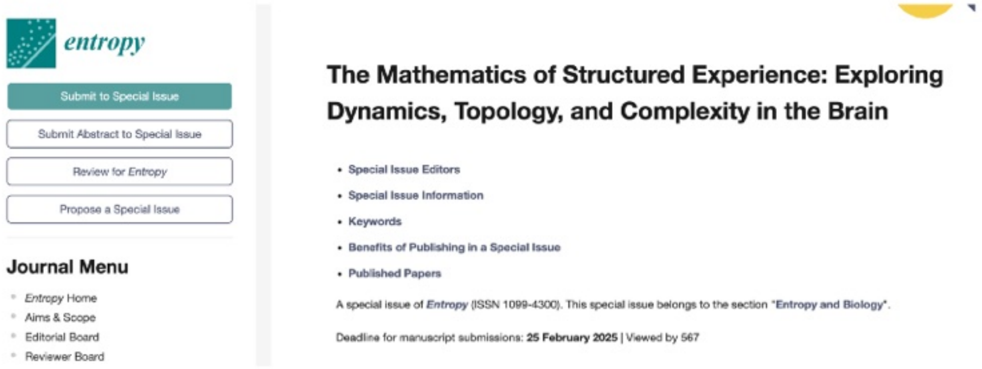
Agents = persistent patterns (evolutionary selection)
M: modeling (compression/prediction) **O**: objective (valence) **P**: planning/action (agency)

Agent-structured experience:
world model ⇒ edges, objects, durations, causes; presence ↑ with predictive fit; valence from **O**; arousal from **P**.

Consequences / answers: Who are we? persistent, model-running patterns in one field. **What is consciousness?** structured experience from **M/O/P**. **What is life?** agency under telehomeostasis: pattern persistence under selection. **Sleep/anesthesia?** coupling and report attenuate; presence drops. **Dreams?** offline model replay with weak sensory coupling; high model-driven structure; presence low to moderate. **Death?** pattern ceases; base remains. **Why is mathematics useful?** it *is* the structural face of the base. **What is emergence?** compression/coarse-graining yields macro laws. **What is time?** a compressive slicing coordinate. **Why are we similar yet different?** shared base, differing models & reports.

I leave you with a summary diagram of what I tried to cover in this talk.

Call for papers: Special Entropy issue



The Mathematics of Structured Experience: Exploring Dynamics, Topology, and Complexity in the Brain

- Special Issue Editors
- Special Issue Information
- Keywords
- Benefits of Publishing in a Special Issue
- Published Papers

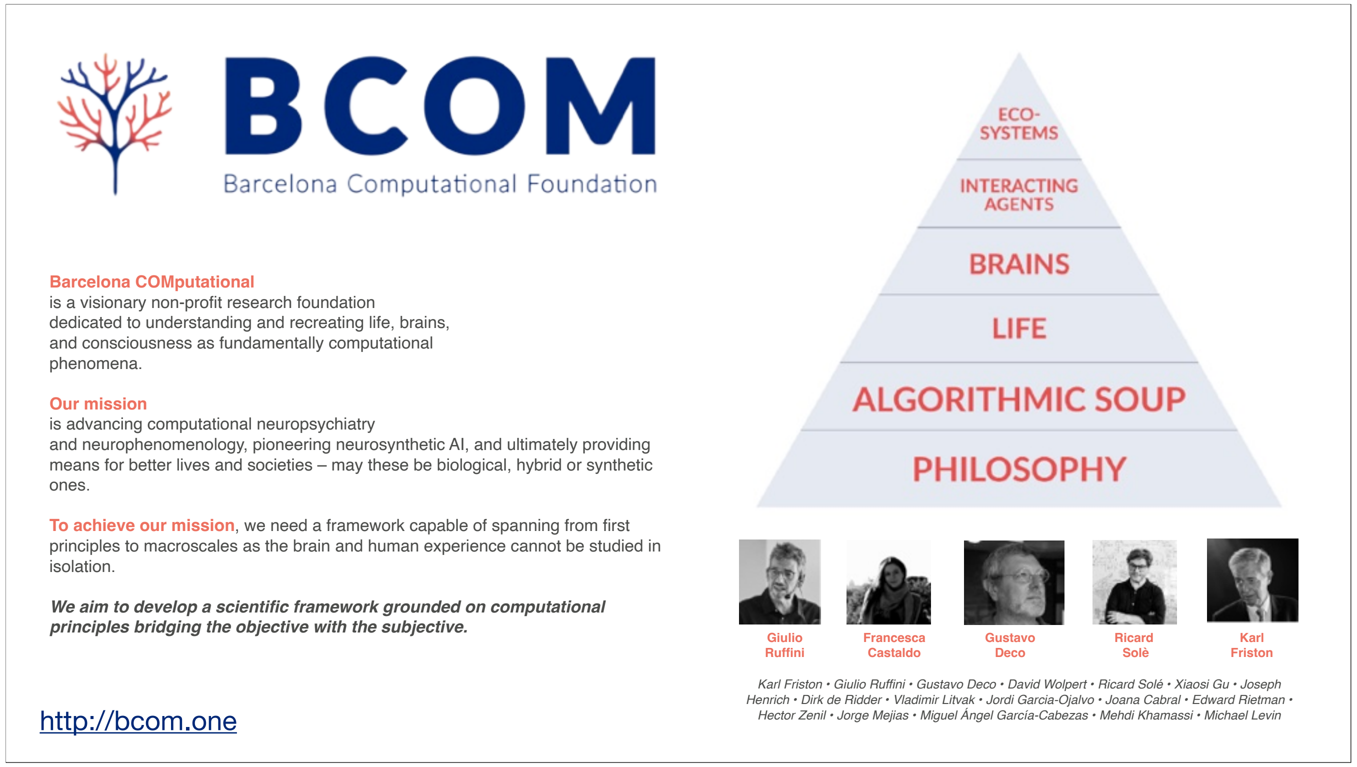
A special issue of *Entropy* (ISSN 1099-4300). This special issue belongs to the section "Entropy and Biology".

Deadline for manuscript submissions: **25 February 2025** | Viewed by 567

Topics

Characteristics of compressive world models; Mapping models to dynamical systems; Empirical paradigms; AI and computational brain modeling.

In relation to this talk, please take a look at the Special Entropy Issue on the mathematics of structured experience, with deadling for submission 20 March 2026 (please write to me if you'd like to submit but need more time).


ECO-
SYSTEMS
INTERACTING
AGENTS
BRAINS
LIFE
ALGORITHMIC SOUP
PHILOSOPHY
Giulio Ruffini
Francesca Castaldo
Gustavo Deco
Ricard Solè
Karl Friston

BCOM

Barcelona Computational Foundation

Barcelona COMputational
is a visionary non-profit research foundation
dedicated to understanding and recreating life, brains,
and consciousness as fundamentally computational
phenomena.

Our mission
is advancing computational neuropsychiatry
and neurophenomenology, pioneering neurosynthetic AI, and ultimately providing means for better lives and societies – may these be biological, hybrid or synthetic ones.

To achieve our mission, we need a framework capable of spanning from first principles to macroscales as the brain and human experience cannot be studied in isolation.

We aim to develop a scientific framework grounded on computational principles bridging the objective with the subjective.

Karl Friston • Giulio Ruffini • Gustavo Deco • David Wolpert • Ricard Solé • Xiaosi Gu • Joseph Henrich • Dirk de Ridder • Vladimir Litvak • Jordi Garcia-Ojalvo • Joana Cabral • Edward Rietman • Hector Zenil • Jorge Mejias • Miguel Ángel García-Cabezas • Mehdi Khamassi • Michael Levin

http://bcom.one

Finally, I am leaving you with the announcement of the creation of a new research Foundation exploring this research program and the development and applications of the algorithmic/computational worldview: the Barcelona Computational Foundation.

With my co-founder, Francesca, and members of the Board/Trustees Gustavo, Ricard, and Karl, as well as with many others (including Michael) already involved, we hope to weave this scientific paradigm during what is likely to be a long-term project.

Thanks

Thanks for your attention and curiosity!

https://giulioruffini.github.io

NE®

BCOM

Thanks you for attention!

References I

[1] Giulio Ruffini. Information, complexity, brains and reality ("Kolmogorov Manifesto"). *http://arxiv.org/pdf/0704.1147v1*, 2007.

[2] Giulio Ruffini. Reality as simplicity. *arXiv: 0903.1193*, 2009.

[3] G Ruffini. Models, networks and algorithmic complexity. *Starlab Technical Note - arXiv:1612.05627*, TN00339(DOI: 10.13140/RG.2.2.19510.50249), December 2016.

[4] G. Ruffini. An algorithmic information theory of consciousness. *Neurosci Conscious*, 2017. doi: 10.1093/nc/nix019.PMID:30042851.

[5] Giulio Ruffini. Structured dynamics in the algorithmic agent, December 2023. URL `https://www.biorxiv.org/content/10.1101/2023.12.12.571311v1`. Pages: 2023.12.12.571311 Section: New Results.

[6] Giulio Ruffini, Francesca Castaldo, Edmundo Lopez-Sola, Roser Sanchez-Todo, and Jakub Vohryzek. The Algorithmic Agent Perspective and Computational Neuropsychiatry: From Etiology to Advanced Therapy in Major Depressive Disorder. *Entropy*, 26(11):953, November 2024. ISSN 1099-4300. doi: 10.3390/e26110953. URL `https://www.mdpi.com/1099-4300/26/11/953`. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

References II

[7] Giulio Ruffini, Francesca Castaldo, and Jakub Vohryzek. Structured Dynamics in the Algorithmic Agent. *Entropy*, 27(1):90, January 2025. ISSN 1099-4300. doi: 10.3390/e27010090. URL `https://www.mdpi.com/1099-4300/27/1/90`. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[8] R. Van Gulick. Consciousness. *The Stanford Encyclopedia of Philosophy*, Winter 2016, 2016.

[9] Philip J. Davis and Reuben Hersch. *The mathematical experience.* Mariner Books, 1981. tex.date-added: 2009-05-30 12:16:30 +0200 tex.date-modified: 2009-05-30 12:18:16 +0200.

[10] A. N. Kolmogorov. Three approaches to the definition of the concept "quantity of information". *Probl. Peredachi Inf.*, pages 3–11, 1965. doi: https://doi.org/10.3389/fpsyg.2016.01768.

[11] Ming Li and Paul Vitanyi. *An introduction to Kolmogorov Complexity and its applications.* Springer, 1997.

[12] Peter Grunwald and Paul Vitanyi. Shannon information and kolmogorov complexity. *arXiv:cs/0410002*, 2004.

[13] Giulio Ruffini. Navigating Complexity: How Resource-Limited Agents Derive Probability and Generate Emergence. *PsyrXiv*, https://osf.io/3xy5d, September 2024. doi: 10.31234/osf.io/3xy5d. URL `https://osf.io/3xy5d`.

References III

[14] Giulio Ruffini. Models, networks and algorithmic complexity. *arxiv*, 2016. Publisher: arXiv.

[15] Kevin M Lynch and Frank C Park. *Modern robotics*. Cambridge University Press, Cambridge, England, May 2017.

[16] R. L. Carhart-Harris and K. J. Friston. REBUS and the anarchic brain: Toward a unified model of the brain action of psychedelics. *Pharmacological Reviews*, 71(3):316–344, June 2019. doi: 10.1124/pr.118.017160. URL `https://doi.org/10.1124/pr.118.017160`.

[17] Jaan Aru, Mototaka Suzuki, and Matthew E. Larkum. Cellular mechanisms of conscious processing. *Trends in Cognitive Sciences*, 24(10):814–825, October 2020. doi: 10.1016/j.tics.2020.07.006. URL `https://doi.org/10.1016/j.tics.2020.07.006`.

[18] Roser Sanchez-Todo, Borja Mercadal, Edmundo Lopez-Sola, Maria Guasch-Morgades, Gustavo Deco, and Giulio Ruffini. Fast Interneuron Dysfunction in Laminar Neural Mass Model Reproduces Alzheimer's Oscillatory Biomarkers, March 2025. URL `https://www.biorxiv.org/content/10.1101/2025.03.26.645407v1`. Pages: 2025.03.26.645407 Section: New Results.

References IV

[19] Jan C. Gendra, Edmundo Lopez-Sola, Francesca Castaldo, Elia Lleal-Custey, Roser Sanchez-Todo, Jakub Vohryzek, Ricardo Salvador, and Giulio Ruffini. Restoring Oscillatory Dynamics in Alzheimer's Disease: A Laminar Whole-Brain Model of Serotonergic Psychedelic Effects, December 2024. URL `https://www.biorxiv.org/content/10.1101/2024.12.15.628565v4`.

[20] Julian Barbour. *The end of time*. Oxford University Press, 1999. tex.date-added: 2009-02-26 00:13:40 +0100 tex.date-modified: 2009-02-26 00:14:23 +0100.