

Category Theory in Science and Engineering

David I. Spivak

(AFOSR grants FA9550-20-1-0348 and FA9550-23-1-0376)



Caltech, Special Seminar in Mechanical and Civil Engineering
2023 December 14

Outline

1 Introduction

- Mathematics as accounting
- Plan for the talk

2 Operads: a framework for compositional operations

3 The algebra of dynamical systems

4 Dynamic operads: where arrangements can change

5 Conclusion

Why am I here?

I've loved **math** and somehow knew it was **powerful** since I was very young.

- I drew **lightning bolts** on my **math notebook** at 6 years old.
- I was trying to figure out how it all fit together and with the world.

Why am I here?

I've loved **math** and somehow knew it was **powerful** since I was very young.

- I drew **lightning bolts** on my **math notebook** at 6 years old.
- I was trying to figure out how it all fit together and with the world.

In 2007 I got my PhD in math at UC Berkeley, on “derived manifolds”.

- It was a geometry result, but **category theory** did all the heavy lifting.
- I found CT beautiful, because it **compressed and organized** so much.

Why am I here?

I've loved **math** and somehow knew it was **powerful** since I was very young.

- I drew **lightning bolts** on my **math notebook** at 6 years old.
- I was trying to figure out how it all fit together and with the world.

In 2007 I got my PhD in math at UC Berkeley, on “derived manifolds”.

- It was a geometry result, but **category theory** did all the heavy lifting.
- I found CT beautiful, because it **compressed and organized** so much.

During my UO postdoc, I realized CT could be much more broadly applied.

- I ambitiously wanted to use it to organize all of human knowledge.
- My work on categ'al databases led to a 10-year stint at MIT in math.

Why am I here?

I've loved **math** and somehow knew it was **powerful** since I was very young.

- I drew **lightning bolts** on my **math notebook** at 6 years old.
- I was trying to figure out how it all fit together and with the world.

In 2007 I got my PhD in math at UC Berkeley, on “derived manifolds”.

- It was a geometry result, but **category theory** did all the heavy lifting.
- I found CT beautiful, because it **compressed and organized** so much.

During my UO postdoc, I realized CT could be much more broadly applied.

- I ambitiously wanted to use it to organize all of human knowledge.
- My work on categ'al databases led to a 10-year stint at MIT in math.

But where were the **lightning bolts**? Where does the power come from?

- I started thinking about dynamical systems working together.
- A NASA grant led to a workshop with Aaron Ames and Paulo Tabuada
- I'm working with them to understand their functorial Lyapunov theory.

I'm here because I'm still trying to understand how **math's language works**.

Mathematical fields as accounting systems

I think of mathematical fields as **well-ruled accounting systems**.

- Arithmetic accounts for the flow of quantities, as in finance.
- Hilbert spaces account for the states of elementary particles, as in QM.
- Probability distributions account for likelihoods, as in game theory.
- Calculus accounts for relative rates of change.
- They're all well-ruled in the sense of being structured and lawful.

Mathematical fields as accounting systems

I think of mathematical fields as **well-ruled accounting systems**.

- Arithmetic accounts for the flow of quantities, as in finance.
- Hilbert spaces account for the states of elementary particles, as in QM.
- Probability distributions account for likelihoods, as in game theory.
- Calculus accounts for relative rates of change.
- They're all well-ruled in the sense of being structured and lawful.

We want **systematic accounting** for what makes things **work**.

- Math'ns like Newton, Pascal, Frege made new accounting systems.
- Carefully **track** the phenomena, **articulate** the structure, **systematize**.
- We need systematic accounting throughout science and engineering.

Mathematical fields as accounting systems

I think of mathematical fields as **well-ruled accounting systems**.

- Arithmetic accounts for the flow of quantities, as in finance.
- Hilbert spaces account for the states of elementary particles, as in QM.
- Probability distributions account for likelihoods, as in game theory.
- Calculus accounts for relative rates of change.
- They're all well-ruled in the sense of being structured and lawful.

We want **systematic accounting** for what makes things **work**.

- Math'ns like Newton, Pascal, Frege made new accounting systems.
- Carefully **track** the phenomena, **articulate** the structure, **systematize**.
- We need systematic accounting throughout science and engineering.

Category theory (CT) is the accounting system for interlocking structures.

- Mathematical definitions are composed of interlocking structures.
- Category theory tracks the layers of structure and their connections.
- This makes analogies—similarities of structure—into formal objects.
- It accounts for the fact that different accounting systems cohere.

Category theory in science and engineering

Over the past 15 years, I've worked with many scientists and engineers, e.g.:

- Amgen, J&J: data modeling and transformations,
- Buehler's lab at MIT CEE: hierarchical protein materials,
- NASA grant: temporal type theory for safe separation of airplanes,
- A. Censi: collaborative design of robots, transportation systems, etc.,
- Ames & Tabuada: their functorial Lyapunov theory.

Category theory in science and engineering

Over the past 15 years, I've worked with many scientists and engineers, e.g.:

- Amgen, J&J: data modeling and transformations,
- Buehler's lab at MIT CEE: hierarchical protein materials,
- NASA grant: temporal type theory for safe separation of airplanes,
- A. Censi: collaborative design of robots, transportation systems, etc.,
- Ames & Tabuada: their functorial Lyapunov theory.

In each case, CT articulates the big themes in precise mathematics.

- Structural relations between different aspects are recorded.
- Single notational, terminological, methodological system...
- ...across a wide array of different examples, or systems of examples.

Category theory in science and engineering

Over the past 15 years, I've worked with many scientists and engineers, e.g.:

- Amgen, J&J: data modeling and transformations,
- Buehler's lab at MIT CEE: hierarchical protein materials,
- NASA grant: temporal type theory for safe separation of airplanes,
- A. Censi: collaborative design of robots, transportation systems, etc.,
- Ames & Tabuada: their functorial Lyapunov theory.

In each case, CT articulates the big themes in precise mathematics.

- Structural relations between different aspects are recorded.
- Single notational, terminological, methodological system...
- ...across a wide array of different examples, or systems of examples.

Category theory is like a conceptual stem cell.

- A stem cell can differentiate into huge variety of forms.
- Coming from a common origin, these forms work together coherently.

Category theory as conceptual stem-cell

Category theory (CT) can differentiate into many forms:

- All forms of pure math... (algebra, topology, logic, number theory, differential geometry...)

Category theory as conceptual stem-cell

Category theory (CT) can differentiate into many forms:

- All forms of pure math... (algebra, topology, logic, number theory, differential geometry...)
- Databases and knowledge representation (categories and functors)
- Functional programming languages (cartesian closed categories)
- Dynamical systems and fractals (operad-algebras, co-algebras)
- Shannon Entropy (operad of simplices, internal algebras)
- Taxonomies, metric spaces, and networks (enriched categories)
- Measurements of diversity in populations (magnitude of categories)
- Open economic game theory (lens categories)
- Collaborative design (enriched categories and profunctors)
- Stability and control (monoidal and enriched categories)
- Petri nets and chemical reaction networks (monoidal categories)
- Quantum processes and NLP (compact closed categories)
- Disease modeling and compartmental models (hypergraph categories)
- Deep learning and prediction markets (dynamic monoidal categories)

Popper's objection

“A theory that explains everything explains nothing.” – Karl Popper

Popper's objection

“A theory that explains everything explains nothing.” – Karl Popper

We counter this objection in two ways:

- Couldn't the same objection be made about mathematics?
 - Mathematics is the basis of hard science, used everywhere.
 - CT—like math—explains, models, formalizes many many things.
 - Conclude that math/CT explains everything and hence nothing?

Popper's objection

“A theory that explains everything explains nothing.” – Karl Popper

We counter this objection in two ways:

- Couldn't the same objection be made about mathematics?
 - Mathematics is the basis of hard science, used everywhere.
 - CT—like math—explains, models, formalizes many many things.
 - Conclude that math/CT explains everything and hence nothing?
- Stem cells don't do work until they differentiate.
 - “Adult-level” work requires differentiation and optimization.
 - But the unified origins lead to impressive interoperability.
 - This helps sciencists/eng'rs bring disparate areas of math to bear.

CT is the gateway to pure mathematics

CT is humanity's most powerful thought-compression language.

CT is the gateway to pure mathematics

CT is humanity's most powerful thought-compression language.

- Designed to transport theorems from one area of math to another.
 - Example: from topology (shapes) to algebra (equations).
 - This isn't mere analogy, it's analogy made rigorous.

CT is the gateway to pure mathematics

CT is humanity's most powerful thought-compression language.

- Designed to transport theorems from one area of math to another.
 - Example: from topology (shapes) to algebra (equations).
 - This isn't mere analogy, it's analogy made rigorous.
- It's revolutionized pure math since its inception in 1940s.
 - It's touched or greatly influenced all corners of mathematics.
 - It's become a gateway to learning mathematics.

CT is the gateway to pure mathematics

CT is humanity's most powerful thought-compression language.

- Designed to transport theorems from one area of math to another.
 - Example: from topology (shapes) to algebra (equations).
 - This isn't mere analogy, it's analogy made rigorous.
- It's revolutionized pure math since its inception in 1940s.
 - It's touched or greatly influenced all corners of mathematics.
 - It's become a gateway to learning mathematics.
- And it's branched out from math in a big way.
 - Databases and knowledge representation ([categories and functors](#))
 - Functional programming languages ([cartesian closed categories](#))
 - Dynamical systems and fractals ([operad-algebras, co-algebras](#))
 - Shannon Entropy ([operad of simplices, internal algebras](#))
 - Taxonomies, metric spaces, and networks ([enriched categories](#))
 - Measurements of diversity in populations ([magnitude of categories](#))
 - Open economic game theory ([lens categories](#))
 - Collaborative design ([enriched categories and profunctors](#))
 - Stability and control ([monoidal and enriched categories](#))
 - Petri nets and chemical reaction networks ([monoidal categories](#))
 - Quantum processes and NLP ([compact closed categories](#))
 - Disease modeling and compartmental models ([hypergraph categories](#))
 - Deep learning and prediction markets ([dynamic monoidal categories](#))

Plan of the talk

Overarching idea: CT as math to scaffold accounts from many disciplines.

- To pick one, we'll discuss *operads*: compositional arrangements.
- I'll sketch a definition and give a lot of examples.
- I'll explain dynamic operads, where arrangements change in time.
 - Examples: ANNs, prediction markets, and compositional physics.
- I'll conclude with a summary.

Outline

- 1 Introduction
- 2 Operads: a framework for compositional operations**
 - Operads: e pluribus unum
 - Examples of operads
 - Summary on operads
- 3 The algebra of dynamical systems
- 4 Dynamic operads: where arrangements can change
- 5 Conclusion

What are compositional operations?

Operations take **arrangements** of many **sorts** and produce one **sort**.

What are compositional operations?

Operations take **arrangements** of many **sorts** and produce one **sort**.

An operad consists of:

- A collection of **sorts** X, Y, \dots ,
- And ways to **arrange** them, $\varphi: X_1, \dots, X_k \rightarrow Y$,
- Such that arrangements can be **nested** inside each other.

What are compositional operations?

Operations take **arrangements** of many **sorts** and produce one **sort**.

An operad consists of:

- A collection of **sorts** X, Y, \dots ,
- And ways to **arrange** them, $\varphi: X_1, \dots, X_k \rightarrow Y$,
- Such that arrangements can be **nested** inside each other.
(That last part is the compositionality.)

What are compositional operations?

Operations take **arrangements** of many **sorts** and produce one **sort**.

An operad consists of:

- A collection of **sorts** X, Y, \dots ,
- And ways to **arrange** them, $\varphi: X_1, \dots, X_k \rightarrow Y$,
- Such that arrangements can be **nested** inside each other.
(That last part is the compositionality.)

Slightly more formal definition to come.

Operads are everywhere

Operads are used unconsciously in many fields.

- Electrical engineering: “wiring diagrams”
- Design: “set-based design”
- Computer programming: “data flow”
- Natural language processing: “grammars”
- Materials science: “hierarchical materials”
- Information theory: “Shannon entropy”

Operads are everywhere

Operads are used unconsciously in many fields.

- Electrical engineering: “wiring diagrams”
- Design: “set-based design”
- Computer programming: “data flow”
- Natural language processing: “grammars”
- Materials science: “hierarchical materials”
- Information theory: “Shannon entropy”

We want to bring operads to the fore.

- There's a common theme in the way we think.
- Operads structure this sort of thinking.
- With mathematical structure, we can go much further.

Operads are everywhere

Operads are used unconsciously in many fields.

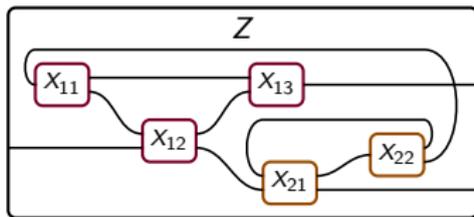
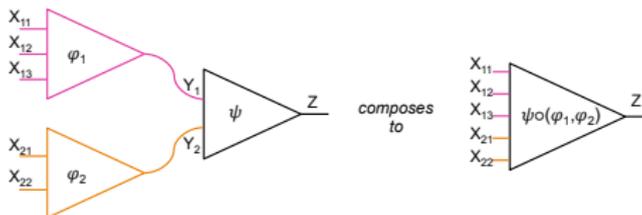
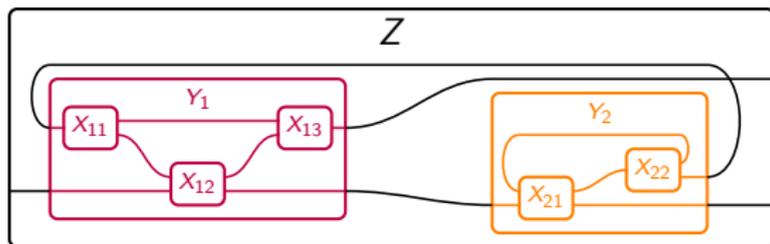
- Electrical engineering: “wiring diagrams”
- Design: “set-based design”
- Computer programming: “data flow”
- Natural language processing: “grammars”
- Materials science: “hierarchical materials”
- Information theory: “Shannon entropy”

We want to bring operads to the fore.

- There's a common theme in the way we think.
- Operads structure this sort of thinking.
- With mathematical structure, we can go much further.

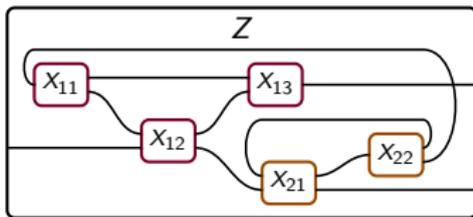
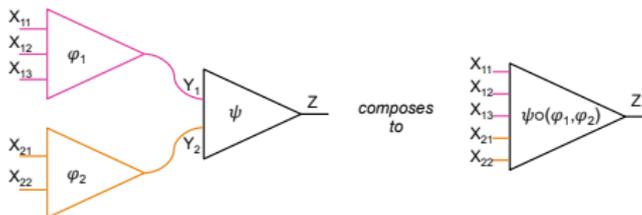
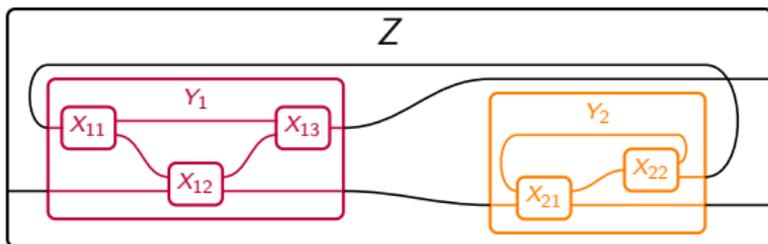
Let's look for **sorts**, **arrangements**, and **nesting** in some examples.

Operad 1: wiring diagrams



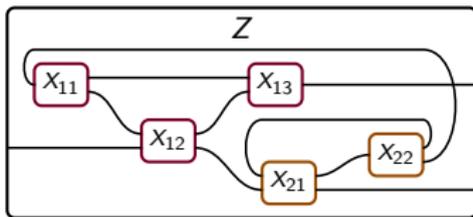
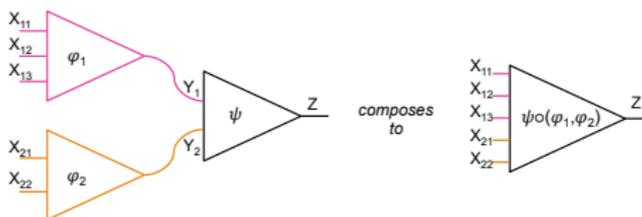
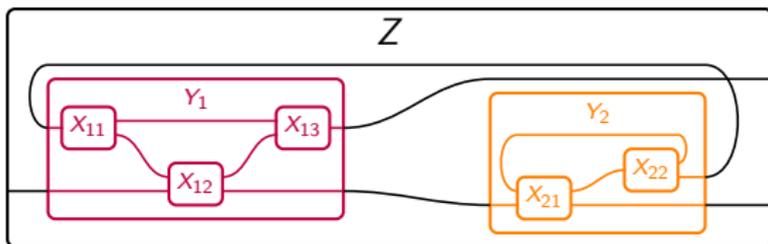
Sorts: boxes with ports.

Operad 1: wiring diagrams



Sorts: boxes with ports. **Arrangements:** wiring diagrams.

Operad 1: wiring diagrams



Sorts: boxes with ports. **Arrangements:** wiring diagrams. **Nesting:** composition.

Formal definition of operad

An operad \mathcal{O} consists of

- A set $\text{Ob}(\mathcal{O})$, elements of which are called *sorts*.
- For sorts $X_1, \dots, X_k, Y \in \text{Ob}(\mathcal{O})$, a set

$$\text{Mor}_{\mathcal{O}}(X_1, \dots, X_k; Y)$$

Its elements are called *morphisms* or *arrangements* of X_1, \dots, X_k in Y .
A k -ary arrangement $\varphi \in \text{Mor}_{\mathcal{O}}(X_1, \dots, X_k; Y)$ may be denoted

$$\varphi: (X_1, \dots, X_k) \rightarrow Y.$$

- For each sort $X \in \text{Ob}(\mathcal{O})$, an identity arrangement $\text{id}_X: (X) \rightarrow X$.
- A composition, or *nesting* formula, e.g.,

$$\psi \circ (\varphi_1, \dots, \varphi_k): (X_{i;j}) \xrightarrow{\varphi_i} (Y_i) \xrightarrow{\psi} Z.$$

These are required to satisfy well-known “unital” and “associative” laws.

Operad 1: WDs again

An operad \mathcal{W} for composing wiring diagrams:

- Sort $X \in \mathcal{W}$: any possible **box-with-ports**.



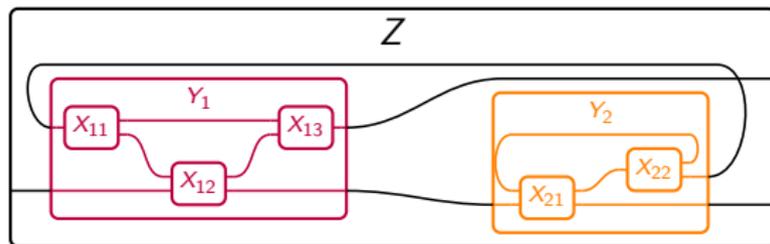
Operad 1: WDs again

An operad \mathcal{W} for composing wiring diagrams:

- Sort $X \in \mathcal{W}$: any possible **box-with-ports**.



- Arrangement $\varphi: X_1, \dots, X_k \rightarrow Y$ in \mathcal{W} : any **wiring** of X 's in Y .
- Nesting: the facts about this **fractal** of wiring possibilities.



- (You could imagine an open dynamical system in each box.)

\mathcal{W} is the decision of what sorts and arrangements you're considering.

Operad 2: hierarchical protein materials

There is an operad \mathcal{M} for composing hierarchical protein materials.

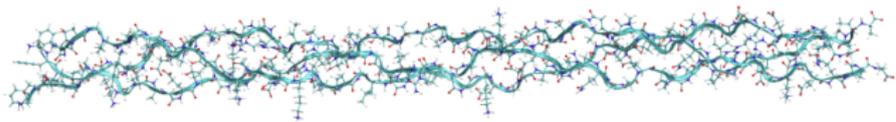
- Why protein materials?
 - Protein materials include your skin: stretchable, breathable, waterproof.
 - Eat hamburgers, make amazing material.
 - Materials scientists would *love* to make materials like this.

¹Giesa, T.; Jagadeesan, R.; Spivak, D.I.; Buehler, M.J. (2015) "Matriarch: a Python library for materials architecture."
ACS Biomaterials Science & Engineering.

Operad 2: hierarchical protein materials

There is an operad \mathcal{M} for composing hierarchical protein materials.

- Why protein materials?
 - Protein materials include your skin: stretchable, breathable, waterproof.
 - Eat hamburgers, make amazing material.
 - Materials scientists would *love* to make materials like this.
- A **protein** is an **arrangement** of simpler **proteins**.
 - There are “atomic” proteins: amino acids.
 - arrange in series or parallel (H-bonds), or
 - arrange in helices, double helices, any conceivable curve, etc.



- Collagen has a **nested** structure: it is an array, each fiber of which is a triple helix, each strand of which is a helix, each unit of which is an amino acid.¹

¹Giesa, T.; Jagadeesan, R.; Spivak, D.I.; Buehler, M.J. (2015) “Matriarch: a Python library for materials architecture.” *ACS Biomaterials Science & Engineering*.

Operad 3: probabilities

Even an operad with **one sort** can be interesting.

- Consider the operad \mathcal{P} for “probabilities”.

Operad 3: probabilities

Even an operad with **one sort** can be interesting.

- Consider the operad \mathcal{P} for “probabilities”.
- Say **sorts**={event}, i.e. “event” is the only sort in \mathcal{P} .
- **Arrangements** = probability distributions. **Nesting** = weighted sums.

Operad 3: probabilities

Even an operad with **one sort** can be interesting.

- Consider the operad \mathcal{P} for “probabilities”.
- Say **sorts** = {event}, i.e. “event” is the only sort in \mathcal{P} .
- **Arrangements** = probability distributions. **Nesting** = weighted sums.
- Formally: $\mathcal{P}_k := \{(x_1, \dots, x_k) \in \mathbb{R}_{\geq 0}^k \mid x_1 + \dots + x_k = 1\}$.

Operad 3: probabilities

Even an operad with **one sort** can be interesting.

- Consider the operad \mathcal{P} for “probabilities”.
- Say **sorts** = {event}, i.e. “event” is the only sort in \mathcal{P} .
- **Arrangements** = probability distributions. **Nesting** = weighted sums.
- Formally: $\mathcal{P}_k := \{(x_1, \dots, x_k) \in \mathbb{R}_{\geq 0}^k \mid x_1 + \dots + x_k = 1\}$.

Arrangement: “In this event, there’s a distribution on next events.”

Operad 3: probabilities

Even an operad with **one sort** can be interesting.

- Consider the operad \mathcal{P} for “probabilities”.
- Say **sorts** = {event}, i.e. “event” is the only sort in \mathcal{P} .
- **Arrangements** = probability distributions. **Nesting** = weighted sums.
- Formally: $\mathcal{P}_k := \{(x_1, \dots, x_k) \in \mathbb{R}_{\geq 0}^k \mid x_1 + \dots + x_k = 1\}$.

Arrangement: “In this event, there’s a distribution on next events.”

- coin flip: $f = (\frac{1}{2}, \frac{1}{2}) \in \mathcal{P}_2$.
 - In the event coin flip, there’s a 50-50 distribution on next events.
- die roll: $r = (\frac{1}{6}, \dots, \frac{1}{6}) \in \mathcal{P}_6$.
- card selection: $p = (\frac{1}{52}, \dots, \frac{1}{52}) \in \mathcal{P}_{52}$.

The **nesting** rule composes distributions by weighted sum:

- Flip a coin: result decides whether to roll a die or pick a card.

$$f \circ (r, p) = \left(\underbrace{\frac{1}{12}, \dots, \frac{1}{12}}_{6 \text{ times}}, \underbrace{\frac{1}{104}, \dots, \frac{1}{104}}_{52 \text{ times}} \right) \in \mathcal{P}_{58}$$

A zoo of operads: Grammars

Any context-free grammar is an operad.

$\langle \text{sentence} \rangle$	$::=$	$\langle \text{noun-phrase} \rangle \langle \text{verb-phrase} \rangle$
$\langle \text{noun-phrase} \rangle$	$::=$	$\langle \text{pronoun} \rangle \mid \langle \text{proper-noun} \rangle \mid \langle \text{determiner} \rangle \langle \text{nominal} \rangle$
$\langle \text{nominal} \rangle$	$::=$	$\langle \text{noun} \rangle \mid \langle \text{nominal} \rangle \langle \text{noun} \rangle$
$\langle \text{verb-phrase} \rangle$	$::=$	$\langle \text{verb} \rangle \mid \langle \text{verb} \rangle \langle \text{noun-phrase} \rangle \mid \langle \text{verb} \rangle \langle \text{prep-phrase} \rangle$
$\langle \text{prep-phrase} \rangle$	$::=$	$\langle \text{preposition} \rangle \langle \text{noun-phrase} \rangle$

How is this an operad?

- The **sorts** are the parts of speech.
- The **arrangements** are the production rules.
- **Nesting** is nesting.

Summary on operads, and a quick word on algebras

Operads show up everywhere.

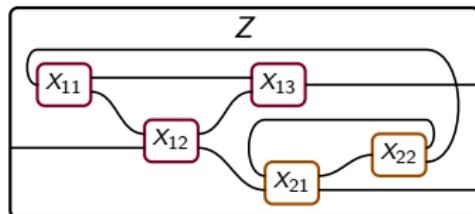
- You can make your own: custom theories of modularity.
- They let you specify how things in your arena of interest are built.

Summary on operads, and a quick word on algebras

Operads show up everywhere.

- You can make your own: custom theories of modularity.
- They let you specify how things in your arena of interest are built.

I didn't talk about algebras, but they're important too.



The operad is syntax, the algebra is semantics: something it can be about.

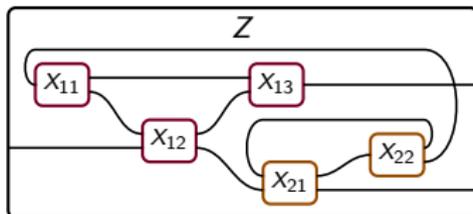
- Imagine filling each of these **boxes** with a little machine.
- Any **arrangement** of machines makes a new machine filling the big box.
- An algebra specifies: the set of things that can fill each **box**, and...
- ...a function from little fillers to big filler, given any **arrangement**.

Summary on operads, and a quick word on algebras

Operads show up everywhere.

- You can make your own: custom theories of modularity.
- They let you specify how things in your arena of interest are built.

I didn't talk about algebras, but they're important too.



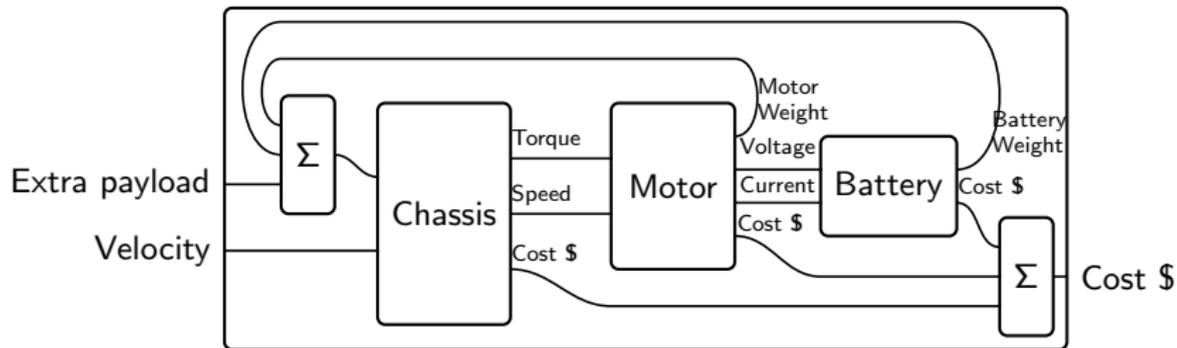
The operad is syntax, the algebra is semantics: something it can be about.

- Imagine filling each of these **boxes** with a little machine.
- Any **arrangement** of machines makes a new machine filling the big box.
- An algebra specifies: the set of things that can fill each **box**, and...
- ...a function from little fillers to big filler, given any **arrangement**.

Next: algebras, then dynamics, then dynamic arrangements.

Andrea Censi's Co-design

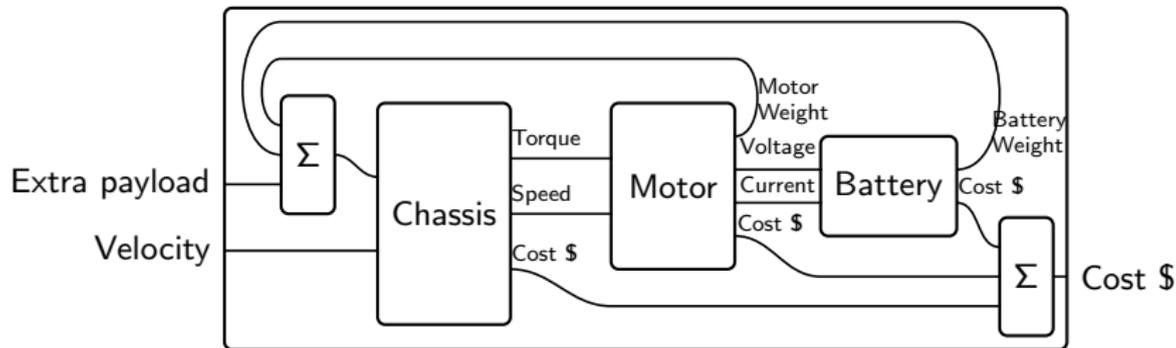
Here is a picture of an **arrangement** in a wiring diagram operad.



- Each wire carries a poset, e.g. of all torques, speeds, weights, costs.

Andrea Censi's Co-design

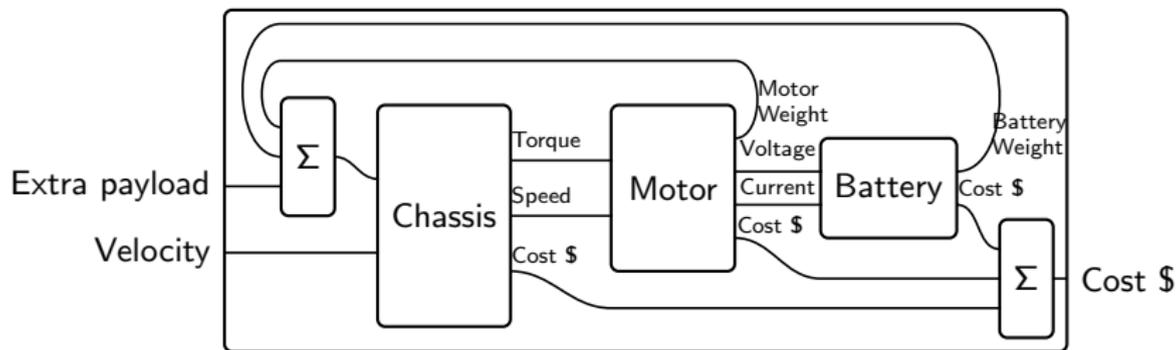
Here is a picture of an **arrangement** in a wiring diagram operad.



- Each wire carries a poset, e.g. of all torques, speeds, weights, costs. Censi defines an algebra, which says what sort of thing can fill each box.
- Each **box** can be filled by a *feasibility* relation $\Phi: A^{\text{op}} \times B \rightarrow \{F, T\}$.
- $\Phi(a, b)$ says if supplying a is feasible given resources b .
 - If $b \leq b'$ and $\Phi(a, b)$ then $\Phi(a, b')$ because “better resources.”
 - If $a' \leq a$ and $\Phi(a, b)$ then $\Phi(a', b)$ because “easier requirements.”

Andrea Censi's Co-design

Here is a picture of an **arrangement** in a wiring diagram operad.



- Each wire carries a poset, e.g. of all torques, speeds, weights, costs.
- Censi defines an algebra, which says what sort of thing can fill each box.
- Each **box** can be filled by a *feasibility* relation $\Phi: A^{\text{op}} \times B \rightarrow \{F, T\}$.
 - $\Phi(a, b)$ says if supplying a is feasible given resources b .
 - If $b \leq b'$ and $\Phi(a, b)$ then $\Phi(a, b')$ because “better resources.”
 - If $a' \leq a$ and $\Phi(a, b)$ then $\Phi(a', b)$ because “easier requirements.”
 - An **arrangement** gives the feas'y of the whole from those of the parts.
 - **Nest** arbitrarily, e.g. each little **box** could contain even littler ones.

Outline

- 1 Introduction
- 2 Operads: a framework for compositional operations
- 3 The algebra of dynamical systems**
 - Manifolds and bundles
 - Lenses and dynamics
 - Interfaces and arrangements of dynamical systems
- 4 Dynamic operads: where arrangements can change
- 5 Conclusion

Manifolds

Manifolds are spaces M that are locally *Euclidean*, i.e. \mathbb{R}^n .

- For example \mathbb{R} , \mathbb{R}^2 , \mathbb{R}^3 , but also $1 := \mathbb{R}^0$ (a point)
- Also unions of these, such as two points, or a circle S^1 or a sphere S^2 .
- Around any point is a little patch that is isomorphic to \mathbb{R}^m .

Manifolds

Manifolds are spaces M that are locally *Euclidean*, i.e. \mathbb{R}^n .

- For example \mathbb{R} , \mathbb{R}^2 , \mathbb{R}^3 , but also $1 := \mathbb{R}^0$ (a point)
- Also unions of these, such as two points, or a circle S^1 or a sphere S^2 .
- Around any point is a little patch that is isomorphic to \mathbb{R}^m .

There are various sorts of maps $f: M \rightarrow N$ that are interesting.

- A map f should send points m in M to points $f(m)$ in N .
- Continuous: send convergent sequences in M to convergent seq's in N .
- Smooth: around any point, the map $\mathbb{R}^m \rightarrow \mathbb{R}^n$ should be smooth.

Manifolds

Manifolds are spaces M that are locally *Euclidean*, i.e. \mathbb{R}^n .

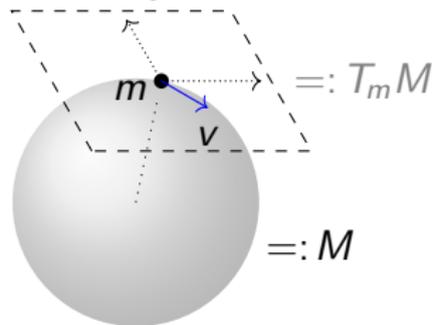
- For example \mathbb{R} , \mathbb{R}^2 , \mathbb{R}^3 , but also $1 := \mathbb{R}^0$ (a point)
- Also unions of these, such as two points, or a circle S^1 or a sphere S^2 .
- Around any point is a little patch that is isomorphic to \mathbb{R}^m .

There are various sorts of maps $f: M \rightarrow N$ that are interesting.

- A map f should send points m in M to points $f(m)$ in N .
- Continuous: send convergent sequences in M to convergent seq's in N .
- Smooth: around any point, the map $\mathbb{R}^m \rightarrow \mathbb{R}^n$ should be smooth.

The (co)tangent bundles TM and T^*M are important for dynamics.

- At each $m \in M$ there's a *tangent space* T_mM .
- T_mM = space of "velocities" of curves thru m .
- It's a vector space: velocities add and scale.
- Its vect-sp dual, T_m^*M , is space of *1-forms*...
- ... (aka *covectors*) at m , measuring velocities.
- TM is union of all T_mM ; similarly for T^*M .



Manifolds

Manifolds are spaces M that are locally *Euclidean*, i.e. \mathbb{R}^n .

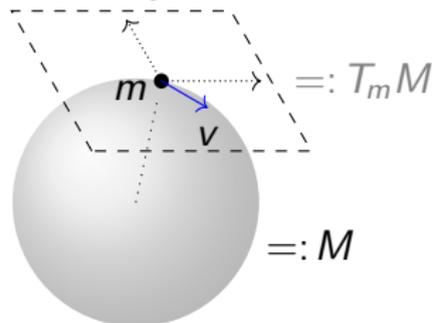
- For example \mathbb{R} , \mathbb{R}^2 , \mathbb{R}^3 , but also $1 := \mathbb{R}^0$ (a point)
- Also unions of these, such as two points, or a circle S^1 or a sphere S^2 .
- Around any point is a little patch that is isomorphic to \mathbb{R}^m .

There are various sorts of maps $f: M \rightarrow N$ that are interesting.

- A map f should send points m in M to points $f(m)$ in N .
- Continuous: send convergent sequences in M to convergent seq's in N .
- Smooth: around any point, the map $\mathbb{R}^m \rightarrow \mathbb{R}^n$ should be smooth.

The (co)tangent bundles TM and T^*M are important for dynamics.

- At each $m \in M$ there's a *tangent space* $T_m M$.
- $T_m M$ = space of "velocities" of curves thru m .
- It's a vector space: velocities add and scale.
- Its vect-sp dual, $T_m^* M$, is space of *1-forms*...
- ... (aka *covectors*) at m , measuring velocities.
- TM is union of all $T_m M$; similarly for T^*M .

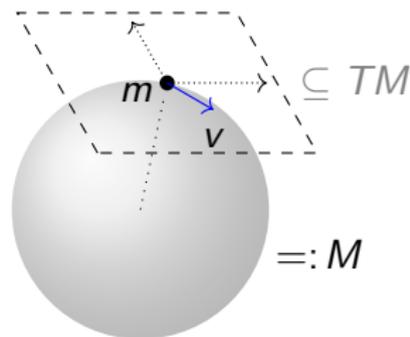


There are proj'n maps $TM \rightarrow M$ and $T^*M \rightarrow M$ sending $(m, v) \mapsto m$.

Bundles

TM and T^*M are bundles; these'll be important.

- A *bundle* over M is a map $\pi: E \rightarrow M$...
- ...with a *local triviality* condition we'll ignore.
- Ex: the proj'ns $TM \rightarrow M$ and $T^*M \rightarrow M$.
- Another example: for any manifold X ,...
- ...the map $X \times M \rightarrow M$ is a bundle.
- *Fiber* over $m \in M$ is T_mM , T_m^*M and X resp.



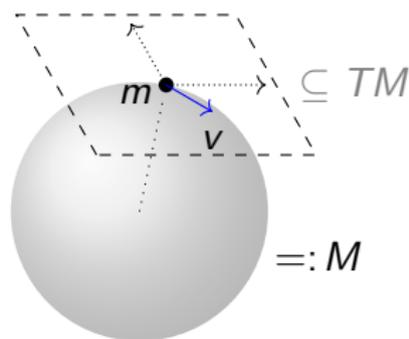
Bundles

TM and T^*M are bundles; these'll be important.

- A *bundle* over M is a map $\pi: E \rightarrow M$...
- ...with a *local triviality* condition we'll ignore.
- Ex: the proj'ns $TM \rightarrow M$ and $T^*M \rightarrow M$.
- Another example: for any manifold X ,...
- ...the map $X \times M \rightarrow M$ is a bundle.
- *Fiber* over $m: M$ is T_mM , T_m^*M and X resp.

Given a map $f: M \rightarrow N$, and a bundle $E \rightarrow N$, get *pullback* $f^*E \rightarrow M$.

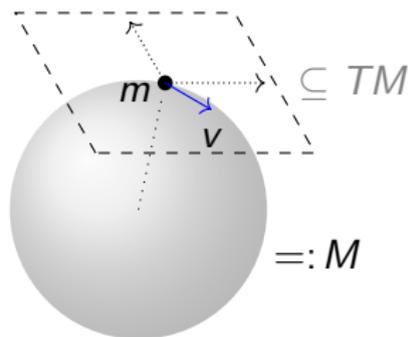
- The fiber of f^*E over each $m: M$ is that of E over $f(m)$.
- So $f^*(TN)$ has the N -tangent space $T_{f(m)}N$ over each $m: M$.



Bundles

TM and T^*M are bundles; these'll be important.

- A *bundle* over M is a map $\pi: E \rightarrow M$...
- ...with a *local triviality* condition we'll ignore.
- Ex: the proj'ns $TM \rightarrow M$ and $T^*M \rightarrow M$.
- Another example: for any manifold X ,...
- ...the map $X \times M \rightarrow M$ is a bundle.
- *Fiber* over $m: M$ is T_mM , T_m^*M and X resp.



Given a map $f: M \rightarrow N$, and a bundle $E \rightarrow N$, get *pullback* $f^*E \rightarrow M$.

- The fiber of f^*E over each $m: M$ is that of E over $f(m)$.
- So $f^*(TN)$ has the N -tangent space $T_{f(m)}N$ over each $m: M$.

The derivative (aka *Jacobian*) of f gives rise to two bundle maps.

- It gives a map $Tf: TM \rightarrow f^*(TN)$ because for every point m ...
- ...the Jacobian gives a linear map $Jf_m: T_mM \rightarrow T_{f(m)}N$.
- But this induces a map $T^*f: f^*(T^*N) \rightarrow T^*M$ because for any m ,...
- ...the map $T_mM \rightarrow T_{f(m)}N$ induces a dual map $T_{f(m)}^*N \rightarrow T_m^*M$.

Lenses and open dynamical systems

We will be more interested in the “backwards” style mapping.

- Say that a *lens* from bundle $E \rightarrow M$ to bundle $E' \rightarrow M'$ is:
 - a pair (f, f^\sharp) , where $f: M \rightarrow M'$ is a smooth map and
 - $f^\sharp: f^*E' \rightarrow E$ is a map on bundles. going the other way.

Notation: we'll denote a bundle $E \rightarrow M$ by $\begin{bmatrix} E \\ M \end{bmatrix}$ and a bundle map

$$\begin{bmatrix} E \\ M \end{bmatrix} \begin{array}{c} \xleftarrow{f^\sharp} \\ \xrightarrow{f} \end{array} \begin{bmatrix} E' \\ M' \end{bmatrix}$$

- Ex: For any smooth $f: M \rightarrow M'$, we get a lens $\begin{bmatrix} T^*M \\ M \end{bmatrix} \begin{array}{c} \xleftarrow{\quad} \\ \xrightarrow{\quad} \end{array} \begin{bmatrix} T^*M' \\ M' \end{bmatrix}$.
- Define the *tensor product* of bundles: $\begin{bmatrix} E_1 \\ B_1 \end{bmatrix} \otimes \begin{bmatrix} E_2 \\ B_2 \end{bmatrix} := \begin{bmatrix} E_1 \times E_2 \\ B_1 \times B_2 \end{bmatrix}$.

Lenses and open dynamical systems

We will be more interested in the “backwards” style mapping.

- Say that a *lens* from bundle $E \rightarrow M$ to bundle $E' \rightarrow M'$ is:
- a pair (f, f^\sharp) , where $f: M \rightarrow M'$ is a smooth map and
- $f^\sharp: f^*E' \rightarrow E$ is a map on bundles. going the other way.

Notation: we'll denote a bundle $E \rightarrow M$ by $\begin{bmatrix} E \\ M \end{bmatrix}$ and a bundle map

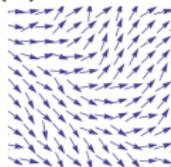
$$\begin{bmatrix} E \\ M \end{bmatrix} \begin{array}{c} \xleftarrow{f^\sharp} \\ \xrightarrow{f} \end{array} \begin{bmatrix} E' \\ M' \end{bmatrix}$$

- Ex: For any smooth $f: M \rightarrow M'$, we get a lens $\begin{bmatrix} T^*M \\ M \end{bmatrix} \begin{array}{c} \xleftarrow{\quad} \\ \xrightarrow{\quad} \end{array} \begin{bmatrix} T^*M' \\ M' \end{bmatrix}$.
- Define the *tensor product* of bundles: $\begin{bmatrix} E_1 \\ B_1 \end{bmatrix} \otimes \begin{bmatrix} E_2 \\ B_2 \end{bmatrix} := \begin{bmatrix} E_1 \times E_2 \\ B_1 \times B_2 \end{bmatrix}$.

We can also encode (parameterized) systems of ODEs this way.

- A lens $\begin{bmatrix} T\mathbb{R}^n \\ \mathbb{R}^n \end{bmatrix} \begin{array}{c} \xleftarrow{\quad} \\ \xrightarrow{\quad} \end{array} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is a pair (f, f^\sharp) where $f: \mathbb{R}^n \rightarrow 1$ (no data) and...
- ... $f^\sharp: f^*1 \rightarrow T\mathbb{R}^n$. What is that? Well $f^*(1) = \mathbb{R}^n$, so...

$$\begin{array}{ccc} \mathbb{R}^n & \xrightarrow{f^\sharp} & T\mathbb{R}^n \\ \parallel & & \swarrow \pi \\ \mathbb{R}^n & & \end{array}$$



That is, it's a vector field on \mathbb{R}^n , i.e. an n -dim'l system of ODEs.

Parameters, readouts, and wiring diagram arrangements

Recall the notion of operad: it consists of interfaces and arrangements.

- What is a reasonable **interface (sort)** $U \dashv \square \dashv R$ for a dynamical system?
- At any time we want to read out a value and pass in a control param.
- Let U be param' space, R readout sp.: an interface is a bundle $I = \left[\begin{array}{c} U \\ R \end{array} \right]$
- The semantics is that we fill any such interface with $\left[\begin{array}{c} TX \\ X \end{array} \right] \begin{array}{c} \xleftarrow{v} \\ \xrightarrow{r} \end{array} \left[\begin{array}{c} U \\ R \end{array} \right]$.
- It's a "readout" map $r: X \rightarrow R$ and $v: U \rightarrow T_x X$ for each $x: X$.
- So (r, v) consists of a readout and a parameterized vector field.

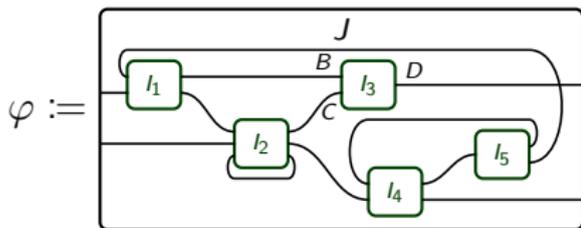
Parameters, readouts, and wiring diagram arrangements

Recall the notion of operad: it consists of interfaces and arrangements.

- What is a reasonable **interface (sort)** $U \dashv \square R$ for a dynamical system?
- At any time we want to read out a value and pass in a control param.
- Let U be param' space, R readout sp.: an interface is a bundle $I = \begin{bmatrix} U \\ R \end{bmatrix}$
- The semantics is that we fill any such interface with $\begin{bmatrix} TX \\ X \end{bmatrix} \xleftrightarrow[r']{v} \begin{bmatrix} U \\ R \end{bmatrix}$.
- It's a "readout" map $r: X \rightarrow R$ and $v: U \rightarrow T_x X$ for each $x: X$.
- So (r, v) consists of a readout and a parameterized vector field.

What are **arrangements**? Later we'll let these be dynamic; for now, static.

- We've seen wiring diagrams before.
- Here interface $I_3 := \begin{bmatrix} B \times C \\ D \end{bmatrix}$, etc.
- It turns out they are all given by...
- ...lens maps $\varphi: I_1 \otimes \cdots \otimes I_5 \xleftrightarrow{\quad} J$.



The WD φ transforms dynamics $\begin{bmatrix} TX_k \\ X_k \end{bmatrix} \xleftrightarrow{\quad} I_k$ in small's to dyn'cs in big.

- Thus dynamical systems form an *algebra* on the WD operad.

Outline

- 1 Introduction
- 2 Operads: a framework for compositional operations
- 3 The algebra of dynamical systems
- 4 Dynamic operads: where arrangements can change**
 - Dynamic operads
 - Artificial neural nets and compositional physics
- 5 Conclusion

Dynamic arrangements

Recall that operads were systems for **nestable arrangements**, e.g.:

- Wiring diagrams (WDs), protein materials, probability distributions, ...
- ... grammars, recipes, etc.

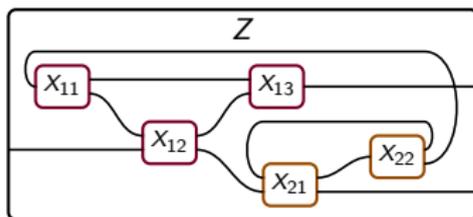
Dynamic arrangements

Recall that operads were systems for **nestable arrangements**, e.g.:

- Wiring diagrams (WDs), protein materials, probability distributions, ...
- ... grammars, recipes, etc.

In each case, what would it mean for the **arrangements** to be dynamic?

- WDs that rewire based on what flows along the wires.



- Protein mat'ls who's configuration changes based on how it's pulled.
- Probability dist'ns that change based on their predictive accuracy.
- Grammatical constructions that change based on commun've success.
- And so on.

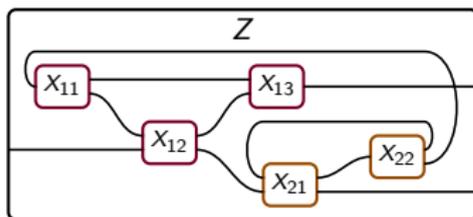
Dynamic arrangements

Recall that operads were systems for **nestable arrangements**, e.g.:

- Wiring diagrams (WDs), protein materials, probability distributions, ...
- ... grammars, recipes, etc.

In each case, what would it mean for the **arrangements** to be dynamic?

- WDs that rewire based on what flows along the wires.



- Protein mat'ls who's configuration changes based on how it's pulled.
- Probability dist'ns that change based on their predictive accuracy.
- Grammatical constructions that change based on commun've success.
- And so on.

The state controls the **arrangement**, and it changes based on what occurs.

Dynamic operads (Joint with Brandon Shapiro)

A *dynamic operad*² is a coherent system of dynamic **arrangements**.

- The coherence means: the changing **arrangements** **nest** lawfully.
- Ex's: ANNs, Hebbian learning, prediction markets, compos'l physics...

²Shapiro, B.T.; Spivak, D.I. "Dynamic operads, dynamic categories: From deep learning to prediction markets".

³Libkind, S.; Spivak, D.I. "A dynamical monoidal category of Hebbian learners"

⁴Work in progress with Owen Lynch and Matteo Capucci

Dynamic operads (Joint with Brandon Shapiro)

A *dynamic operad*² is a coherent system of dynamic **arrangements**.

- The coherence means: the changing **arrangements** **nest** lawfully.
- Ex's: ANNs, Hebbian learning, prediction markets, compos'l physics...

What does the coherence mean in each case?

- Artificial neural networks: basically the chain rule.
- Hebbian learning:³ populations at different scales wire together...
- ...when they fire together, and it doesn't matter how you chunk it.
- Prediction markets: updating by Bayes rule is compositional.

²Shapiro, B.T.; Spivak, D.I. "Dynamic operads, dynamic categories: From deep learning to prediction markets".

³Libkind, S.; Spivak, D.I. "A dynamical monoidal category of Hebbian learners"

⁴Work in progress with Owen Lynch and Matteo Capucci

Dynamic operads (Joint with Brandon Shapiro)

A *dynamic operad*² is a coherent system of dynamic **arrangements**.

- The coherence means: the changing **arrangements** **nest** lawfully.
- Ex's: ANNs, Hebbian learning, prediction markets, compos'l physics...

What does the coherence mean in each case?

- Artificial neural networks: basically the chain rule.
- Hebbian learning:³ populations at different scales wire together...
- ...when they fire together, and it doesn't matter how you chunk it.
- Prediction markets: updating by Bayes rule is compositional.

Compositional physics:⁴

- Idea: let physical systems interact only through their interfaces.
- Each system outputs its momentum and inputs forces.
- These can be arbitrarily nested, e.g. n -tuple pendula.

²Shapiro, B.T.; Spivak, D.I. "Dynamic operads, dynamic categories: From deep learning to prediction markets".

³Libkind, S.; Spivak, D.I. "A dynamical monoidal category of Hebbian learners"

⁴Work in progress with Owen Lynch and Matteo Capucci

Dynamic structures: From ANNs to compositional physics

A static arrangement of interfaces $I_1 \dots I_k$ in J is a lens $I_1 \otimes \dots \otimes I_k \begin{matrix} \xleftarrow{f^\#} \\ \xrightarrow{f} \end{matrix} J$.

- What if we want it to be dynamic? Just tensor on $\left[\begin{matrix} TA \\ A \end{matrix} \right]$ for some f A .
- A is the space of arrangements; if $A = 1$ we get back static arr'nts.

Dynamic structures: From ANNs to compositional physics

A static arrangement of interfaces $I_1 \dots I_k$ in J is a lens $I_1 \otimes \dots \otimes I_k \xrightleftharpoons[f]{f^\#} J$.

- What if we want it to be dynamic? Just tensor on $\begin{bmatrix} TA \\ A \end{bmatrix}$ for some f and A .
- A is the space of arrangements; if $A = 1$ we get back static arr'nts.

In both artificial neural networks and compositional physics, this has form:

$$\begin{bmatrix} TA \\ A \end{bmatrix} \otimes \begin{bmatrix} T^*X_1 \\ X_1 \end{bmatrix} \otimes \dots \otimes \begin{bmatrix} T^*X_k \\ X_k \end{bmatrix} \xrightleftharpoons[(f,H)]{f^\#} \begin{bmatrix} T^*(Y \times \mathbb{R}) \\ Y \times \mathbb{R} \end{bmatrix}$$

These maps form an algebra, and each is induced by three pieces of data:

- *parameterized function* $f: A \times X_1 \times \dots \times X_k \rightarrow Y$,
- *energy map* $H: A \times X_1 \times \dots \times X_k \rightarrow \mathbb{R}$: ANNs $\mapsto 0$; Phys \mapsto Hamiltonian
- *response map* $T^*A \rightarrow TA$: ANNs \mapsto Riemannian; Phys \mapsto symplectic.

Dynamic structures: From ANNs to compositional physics

A static arrangement of interfaces $I_1 \dots I_k$ in J is a lens $I_1 \otimes \dots \otimes I_k \xrightleftharpoons[f]{f^\#} J$.

- What if we want it to be dynamic? Just tensor on $\left[\begin{smallmatrix} TA \\ A \end{smallmatrix} \right]$ for some f A .
- A is the space of arrangements; if $A = 1$ we get back static arr'n'ts.

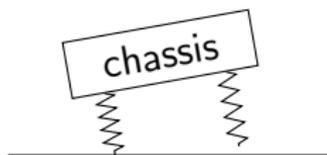
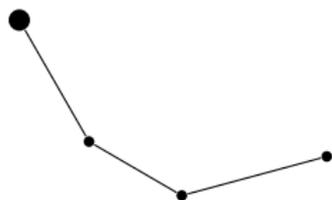
In both artificial neural networks and compositional physics, this has form:

$$\left[\begin{smallmatrix} TA \\ A \end{smallmatrix} \right] \otimes \left[\begin{smallmatrix} T^*X_1 \\ X_1 \end{smallmatrix} \right] \otimes \dots \otimes \left[\begin{smallmatrix} T^*X_k \\ X_k \end{smallmatrix} \right] \xrightleftharpoons[(f, H)]{f^\#} \left[\begin{smallmatrix} T^*(Y \times \mathbb{R}) \\ Y \times \mathbb{R} \end{smallmatrix} \right]$$

These maps form an algebra, and each is induced by three pieces of data:

- *parameterized function* $f: A \times X_1 \times \dots \times X_k \rightarrow Y$,
- *energy map* $H: A \times X_1 \times \dots \times X_k \rightarrow \mathbb{R}$: ANNs $\mapsto 0$; Phys \mapsto Hamiltonian
- *response map* $T^*A \rightarrow TA$: ANNs \mapsto Riemannian; Phys \mapsto symplectic.

Examples in compositional physics: n -tuple pendula, chassis on springs.



Technical slide recapitulating compositional physics

We can describe such dynamic arrangements **DA** by first defining **P**.

- Its objects are **Mfd**-bundles, $\text{Ob } \mathbf{P} = \{\pi: E \rightarrow B\}$, denoted $\left[\begin{smallmatrix} E \\ B \end{smallmatrix} \right]$.
- Morphisms in **P** are lenses as above: $(f: B \rightarrow B', f^\#: f^*\pi' \rightarrow \pi)$.
- Monoidal structure $(\left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix} \right], \otimes)$ by $\left(\left[\begin{smallmatrix} E \\ B \end{smallmatrix} \right] \otimes \left[\begin{smallmatrix} E' \\ B' \end{smallmatrix} \right] \right) := \left[\begin{smallmatrix} E \times E' \\ B \times B' \end{smallmatrix} \right]$
- There's a strong monoidal functor $t^*: \mathbf{Mfd} \rightarrow \mathbf{P}$ given by $M \mapsto \left[\begin{smallmatrix} T^*M \\ M \end{smallmatrix} \right]$
- The \times -mon. $(1 \xrightarrow{0} \mathbb{R}, (+): \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R})$ induces a \otimes -mon. on $r := \left[\begin{smallmatrix} T^*\mathbb{R} \\ \mathbb{R} \end{smallmatrix} \right]$
- The Kleisli cat'y $\mathbf{P}_{\mathbb{R} \times -}$ of this qua monad has maps $\left[\begin{smallmatrix} E \\ B \end{smallmatrix} \right] \leftarrow \left[\begin{smallmatrix} E' \times T^*\mathbb{R} \\ B' \times \mathbb{R} \end{smallmatrix} \right]$
- Define **DA** by the “para construction”: a map $X \rightarrow Y$ in **DA** is...
- ...a choice of $A \in \mathbf{Mfd}$ and map $\left[\begin{smallmatrix} T^*A \\ A \end{smallmatrix} \right] \otimes X \rightarrow Y$ in $\mathbf{P}_{\mathbb{R} \times -}$.

Let's define the multiple-pendulum: e.g. double pendulum:

- We'll do so by def'g a single pendulum as an endomap on $c := \left[\begin{smallmatrix} T^*\mathbb{C} \\ \mathbb{C} \end{smallmatrix} \right]$
- With $a := \left[\begin{smallmatrix} TT^*S^1 \\ T^*S^1 \end{smallmatrix} \right]$ as states, regard $a \otimes c \rightarrow c \otimes r$ as in **DA**(c, c).
- Self-composing n times defines the dynamics of the n -tuple pendulum.

Outline

- 1 Introduction
- 2 Operads: a framework for compositional operations
- 3 The algebra of dynamical systems
- 4 Dynamic operads: where arrangements can change
- 5 **Conclusion**

Summary

Science and engineering involve many models that need to interact.

- Different areas of math account for different types of phenomena.
- We intuitively translate between them to solve bigger problems.

Summary

Science and engineering involve many models that need to interact.

- Different areas of math account for different types of phenomena.
- We intuitively translate between them to solve bigger problems.

Category theory is the math of layered structures and how they relate.

- It's powerful enough to solve long-standing conjectures.
- But it's also very applicable, because we see structure everywhere.

Summary

Science and engineering involve many models that need to interact.

- Different areas of math account for different types of phenomena.
- We intuitively translate between them to solve bigger problems.

Category theory is the math of layered structures and how they relate.

- It's powerful enough to solve long-standing conjectures.
- But it's also very applicable, because we see structure everywhere.

Operads are one example: building new **things** by **arranging** old ones.

- Wiring diagrams of dynamical systems, protein materials, etc...
- Dynamic operads allow arrangements to change as things occur.
- ANNs, prediction markets, Hebbian learning, and...
- ... compositional physics, all fit into this framework.

Summary

Science and engineering involve many models that need to interact.

- Different areas of math account for different types of phenomena.
- We intuitively translate between them to solve bigger problems.

Category theory is the math of layered structures and how they relate.

- It's powerful enough to solve long-standing conjectures.
- But it's also very applicable, because we see structure everywhere.

Operads are one example: building new **things** by **arranging** old ones.

- Wiring diagrams of dynamical systems, protein materials, etc...
- Dynamic operads allow arrangements to change as things occur.
- ANNs, prediction markets, Hebbian learning, and...
- ... compositional physics, all fit into this framework.

Thanks; comments and questions welcome!