

# Dynamic categories, dynamic operads: From deep learning to prediction markets

Brandon Shapiro

David I. Spivak

## Abstract

Natural organized systems adapt to internal and external pressures and this seems to happen all the way down. Wanting to think clearly about this idea motivates our paper, and so the idea is elaborated extensively in the introduction, which should be broadly accessible to a philosophically-interested audience.

In the remaining sections, we turn to more compressed category theory. We define the monoidal double category  $\mathbb{O}\mathbf{rg}$  of dynamic organizations, we provide definitions of  $\mathbb{O}\mathbf{rg}$ -enriched, or *dynamic*, categorical structures—e.g. dynamic categories, operads, and monoidal categories—and we show how they instantiate the motivating philosophical ideas. We give two examples of dynamic categorical structures: prediction markets as a dynamic operad and deep learning as a dynamic monoidal category.

## 1 Introduction

Intuitively, an *open dynamical system* is a machine or worker with an interface by which to interact with whatever else is out there. Open dynamical systems can be organized as circuits or control loops, so that they affect each other by their outward expressions of internal work, and thereby possibly form a more complex worker. The framework here is fractal—or more precisely *operadic*—in its structure: organizations of workers can be nested into arbitrary hierarchies of abstraction.

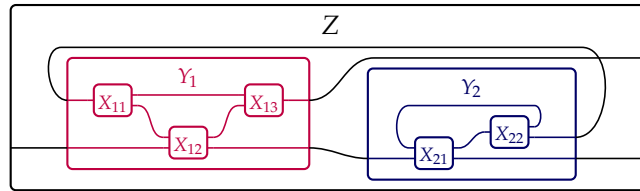


Figure 1: A nesting of interacting open dynamical systems: the  $X_{i,j}$  are wired together to form the  $Y_i$ , which are wired together to form  $Z$ ; often these groupings are chosen to create new abstractions, e.g. in logical circuits or control systems. The permanence of the above-displayed wiring pattern is exactly what is relaxed in this paper; a dynamic organization is one in which interactions may change dynamically based on what flows within the system.

But if we think about some things that interact to do work in the real world, we often notice that often the organization itself—the connections themselves—change. Unlike what we see in Fig. 1, the way we connect this hour may be different from the way we connect next hour; in particular, our interfaces go in and out of contact. At the end of this paragraph, look away from the page for a few seconds, think about some things you know that interact together or influence each other, and ask yourself three questions about them: Do these things ever stop interacting? If so, do they ever start interacting again? And how is it decided?

## 1.1 Accounting for organizational change

We propose that the metaphysical nature and scope of these questions should be complemented by some sort of guard rails to keep our contemplation on track. This is the role of mathematics in our work. It provides a symbolic *accounting system* which is articulate enough to facilitate one in explicating an example and another in asking questions about it.

The category **Poly** of polynomial functors in one variable is an ergonomic mathematical structure with many applications and spin-off categorical gadgets. We will begin in Section 2 by recalling one such gadget from [Spi21]: a category-enriched multicategory **Org** that will be the conceptual centerpiece of our accounting system. Its objects are polynomial functors in one variable, and its morphisms are polynomial coalgebras related to a certain monoidal closed structure on **Poly**. We will see that the morphisms in **Org** are intuitively “collective organizational patterns that change dynamically”.

Leaving the mathematics aside until Section 2—at which point we will have little more to say about the background philosophy—let’s return to the question “how is the organizational pattern between various systems decided, moment-by-moment?” Let’s mesh this question with the idea that the so-organized systems can be nested into arbitrary hierarchies of abstraction. And let’s think about all this in the frame of a certain worldview which we invite you the reader to engage with like a fictional movie, not intended to convince you of fact but instead simply to convey an experience. Here goes.

In this worldview, we notice that everything that makes any sense to us happens to be a collective. A cell body, a human body, an antibody, Topos Institute, an idea, an airport, a sentence, a mathematical definition, a grain of sand, ... each is a collective of interacting parts that may themselves be collectives. Stay with an example or two—e.g. any of the above—for now, not the counterexamples or counterpoints yet, because there are plenty of examples (collectives) and they point more toward the subject-matter of this paper.

It’s quite often the case that these collectives, like the ship of theseus, are not permanent organizations that are fixed for all time; they are adapting to forces from within and without the system. Even a grain of sand can break or melt; even a mathematical definition can be refactored. So then what’s outside the system, generating these forces that influence it? We imagine that what’s outside is in fact more of the same kind of stuff as what’s inside, just not as cohesive perhaps. Let’s go full-on woo: if the universe is a big system, then maybe the sort of thing that happens in our head is—in some way—just like what happens outside of it. Maybe the motives that organize Brandon and David into a collaborative

thinking and paper-writing unit are, in the some reasonable account, of the same nature as the motives that organize each one of them into a body.

But is this right? How could you check such a claim? One would need to give a reasonable account of it, and since we as authors can't currently give such an account, we don't make this claim. Instead, what we present here is an *accounting system* in which the woo-person, (or would it instead be the reductive materialist?) who thought that what went on inside the head was somehow the same as what went on outside, could endeavor to provide such an account of their thinking.

## 1.2 Dynamic categorical structures

Our main definition in this paper is what we call an *dynamic categorical structure*. We might poetically say that a *dynamic* category is one where the morphisms between two objects change in response to what flows between those objects. To define it, we first refactor the definition of  $\mathbb{O}rg$  from [Spi21] from an operad to a monoidal double category; we then define a dynamic *\*thing\** to be a *\*thing\** enriched in  $\mathbb{O}rg$ . Once these are defined, we give a couple examples: a *prediction market* operad and a *deep learning* monoidal category. In the prediction market, a population  $Y$  predicts a distribution based on the predictions of its member populations  $X_i$  weighted by their reputations, and the reputations change dynamically based on the returned outcome. A similar story holds with deep learning.

We thank you the reader for having postponed your counterpoints and counterexamples, and we ask you to reengage both skepticism and interest as you see fit. We invite you to ask openly: what's not a collective of interacting parts that are themselves collectives? Nature, love, or experience perhaps? It all depends on how you look. What we present here is an accounting system for making sense of a certain sort of experiential pattern; the matter itself is whatever it is.

## 1.3 Acknowledgments

The influences on this paper are too numerous and unranked to name, but in particular we thank Sophie Libkind for stimulating conversations, and we thank Scott Garrabrant for teaching us about Kelly betting, which partially inspired Section 4.1.

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0348.

## 2 The Monoidal Double Category $\mathbb{O}rg$

In [Spi21], the second author defined a category-enriched multicategory  $\mathbb{O}rg$ , whose objects are polynomials and whose morphisms are polynomial coalgebras. In this chapter, we describe how  $\mathbb{O}rg$  in fact more naturally takes the form of a monoidal double category, with coalgebras as horizontal morphisms, maps of polynomials as vertical morphisms, and the Dirichlet tensor product  $\otimes$  providing the monoidal structure.<sup>1</sup>

---

<sup>1</sup>In fact,  $\mathbb{O}rg$  is a duoidal double category, with a second monoidal structure given by  $\triangleleft$ , but we will not use that here.

Before we begin, recall that a polynomial is a functor  $p: \mathbf{Set} \rightarrow \mathbf{Set}$  which is isomorphic to a sum of representables; following [Spi21], we denote  $p, q \in \mathbf{Poly}$  by

$$p = \sum_{I \in p(1)} y^{p[I]} \quad \text{and} \quad q = \sum_{J \in q(1)} y^{q[J]} \quad (1)$$

and refer to each  $I \in p(1)$  as a  $p$ -position and to each  $i \in p[I]$  as a  $p$ -direction at  $I$ . A map  $\phi: p \rightarrow q$  of polynomials is a natural transformation. Combinatorially,  $\phi$  provides: for each  $I \in p(1)$  a choice of  $\phi(I) \in q(1)$  and for each  $j \in q[\phi(I)]$  a choice of  $\phi(I, j) \in p[I]$ .<sup>2</sup>

## 2.1 $[p, q]$ -coalgebras

We first recall the definitions of the internal-hom polynomials  $[p, q]$  and concretely describe the category of  $[p, q]$ -coalgebras, which will form the category of morphisms from  $p$  to  $q$  in the underlying bicategory of  $\mathbf{Org}$ .

**Definition 2.1.** For polynomials  $p, q \in \mathbf{Poly}$  as in (1), their *internal hom* with respect to the tensor product  $\otimes$  is the polynomial

$$[p, q] := \sum_{\phi: p \rightarrow q} y^{\sum_{I \in p(1)} q[\phi(I)]} \quad (2)$$

It can also be written  $[p, q] = \prod_{I \in p(1)} \sum_{J \in q(1)} \prod_{j \in q[J]} \sum_{i \in p[I]} 1$ .  $\diamond$

For intuition, a  $[p, q]$ -coalgebra (denoted  $p \rightharpoonup q$ ) is a machine that outputs maps  $\phi: p \rightarrow q$  and that inputs what *flows* between them: pairs  $(I, j)$  where  $I \in p(1)$  is a position of  $p$ , which “flows” to  $q$  as  $J := \phi(I) \in q(1)$ , and  $j \in q[J]$  is a direction of  $q$ , which “flows” backward to  $p$  as  $\phi(I, j) \in p[I]$ . More precisely, using [Spi21, Definition 2.10], we define  $[p, q]$ -coalgebras as follows.

**Definition 2.2.** The category  $[p, q]\text{-Coalg}$  has as objects pairs  $\mathbf{S} = (S, \beta)$  where  $S$  is a set and  $\beta: S \rightarrow [p, q] \triangleleft S$  is a function, and where a morphism from  $\mathbf{S}$  to  $\mathbf{S}'$  is a function  $f: S \rightarrow S'$  making (3) commute.

$$\begin{array}{ccc} S & \xrightarrow{\beta} & [p, q] \triangleleft S \\ f \downarrow & & \downarrow [p, q] \triangleleft f \\ S' & \xrightarrow{\beta'} & [p, q] \triangleleft S' \end{array} \quad (3)$$

We refer to  $S$  as the *state set* and to each element  $s \in S$  as a *state*.  $\diamond$

Unwinding this definition, it is useful to break  $\beta$  into two functions  $\beta := (\text{act}^\beta, \text{upd}^\beta)$ , an *action function*

$$\text{act}^\beta: S \rightarrow \mathbf{Poly}(p, q) = [p, q](1)$$

and, for each state  $s \in S$ , an *update function*

$$\text{upd}_s^\beta: \sum_{I \in p(1)} q[\text{act}_s^\beta(I)] \rightarrow S.$$

---

<sup>2</sup>In [Spi21], what we denote  $\phi(I)$  is denoted  $\phi_1(I)$  and what we denote  $\phi(I, j)$  is denoted  $\phi_1^\#(j)$ .

For a state  $s \in S$  and position  $I \in p(1)$  we often write  $\text{act}_s^\beta: p \rightarrow q$  and  $\text{upd}_s^\beta(I): q[\text{act}_s^\beta(I)] \rightarrow S$ . We may suppress the  $\beta$  when it is clear from context, writing  $\text{act}_s$  and  $\text{upd}_s$ . A coalgebra map  $\mathbf{S} \rightarrow \mathbf{S}'$  is a function  $S \rightarrow S'$  between the state sets that preserves actions and updates.

When, for each  $s \in S$ , the update  $\text{upd}_s$  is the constant function sending everything to  $s$ , we say the coalgebra  $\mathbf{S}$  is *static*, as it remains constantly at  $s$  regardless of the inputs  $I \in p(1)$  and  $j \in q[\text{act}_s(I)]$  flowing between  $p$  and  $q$ .

**Example 2.3.** A special case of a static  $[p, q]$ -coalgebra is given by a map  $\phi \in \mathbf{Poly}(p, q)$ . For each such  $\phi$ , there is a coalgebra  $\{\phi\}$  with a singleton state set and with  $\text{act}^\beta$  sending the point to  $\phi$ ; we call it a *singleton* coalgebra.

A coalgebra is static iff it is the coproduct of singleton coalgebras.  $\diamond$

## 2.2 Composition of hom-coalgebras

We now describe how  $[p, q]$ -coalgebras behave like morphisms from  $p$  to  $q$ .

**Proposition 2.4.** *The categories  $[p, q]$ -Coalg form the hom-categories in a bicategory  $\mathbf{Org}$ , which has polynomials as objects.*

We use  $\mathbf{Org}$  to denote both the bicategory from Proposition 2.4 and the categorical operad in [Spi21, Definition 2.19], as both are derived from the monoidal double category  $\mathbf{Org}$  described in the following sections. For now, we merely present the identities and composites in this bicategory. Identities are easy: the identity object in  $\mathbf{Org}(p, p) = [p, p]$ -Coalg is given by the one-state coalgebra  $\{\text{id}_p\}$ .

The composition functor  $\mathbf{Org}(p, q) \times \mathbf{Org}(q, r) \rightarrow \mathbf{Org}(p, r)$  is defined as the composite:

$$[p, q]\text{-Coalg} \times [q, r]\text{-Coalg} \rightarrow ([p, q] \otimes [q, r])\text{-Coalg} \rightarrow [p, r]\text{-Coalg},$$

where the first functor is the lax monoidality of  $(-)\text{-Coalg}: \mathbf{Poly} \rightarrow \mathbf{Cat}$ , as described in [Spi21, Proposition 2.13], and the second is given by applying  $(-)\text{-Coalg}$  to the usual “composition” map of internal-homs  $[p, q] \otimes [q, r] \rightarrow [p, r]$  in  $\mathbf{Poly}$ . Using (2) we see that on positions, this map takes the form

$$([p, q] \otimes [q, r])(1) = \mathbf{Poly}(p, q) \times \mathbf{Poly}(q, r) \xrightarrow{\circ} \mathbf{Poly}(p, r) = [p, r](1)$$

and on directions it is given for  $\phi: p \rightarrow q$  and  $\psi: q \rightarrow r$  by the function

$$\left( \sum_{I \in p(1)} q[\phi(I)] \right) \times \left( \sum_{J \in q(1)} r[\psi(J)] \right) \leftarrow \sum_{I \in p(1)} r[\psi(\phi(I))]$$

which sends  $(I, k)$  to  $((I, \psi(\phi(I), k)), (\phi(I), j))$ .

Concretely, the composite of a  $[p, q]$ -coalgebra  $\mathbf{S}$  and a  $[q, r]$ -coalgebra  $\mathbf{S}'$  is a  $[p, r]$ -coalgebra which we denote  $\mathbf{S} \circ \mathbf{S}'$  and define as follows:

- its state set is given by  $S \times S'$
- the action of the pair  $(s, s')$  is given by the composite

$$\text{act}_{s, s'}^{\beta \circ \beta'} := (\text{act}_s^\beta \circ \text{act}_{s'}^{\beta'}): p \rightarrow q \rightarrow r$$

- the update function of  $(s, s')$  is induced by the functions

$$\begin{aligned} \sum_{I \in p(1)} r \left[ \text{act}_{s, s'}^{\beta \circ \beta'}(I) \right] &\xrightarrow{(I, k) \mapsto (I, \text{act}_{s'}^{\beta'}(\text{act}_s^\beta(I), k))} \sum_{I \in p(1)} q \left[ \text{act}_s^\beta(I) \right] \xrightarrow{\text{upd}_s^\beta} S, \\ \sum_{I \in p(1)} r \left[ \text{act}_{s, s'}^{\beta \circ \beta'}(I) \right] &\xrightarrow{(I, k) \mapsto (\text{act}_s^\beta(I), k)} \sum_{J \in q(1)} r \left[ \text{act}_{s'}^{\beta'}(J) \right] \xrightarrow{\text{upd}_{s'}^{\beta'}} S'. \end{aligned}$$

Horizontal composition of coalgebra-morphisms—i.e. of the 2-cells in the bicategory  $\mathbb{O}\mathbf{rg}$ —is given simply by the cartesian product. The coherence isomorphisms and axioms for a bicategory then follow from the essential uniqueness of finite products of sets, and the unitality and associativity of composition for polynomial maps.

### 2.3 Monoidal product of coalgebras

It is shown in [Spi21, Proposition 2.13] that the tensor product  $\otimes$  of polynomials extends to make  $\mathbb{O}\mathbf{rg}$  a monoidal bicategory. That is, for polynomials  $p, q, p', q'$  there is a functor

$$[p, q]\text{-}\mathbf{Coalg} \times [p', q']\text{-}\mathbf{Coalg} \rightarrow ([p, q] \otimes [p', q'])\text{-}\mathbf{Coalg} \rightarrow [p \otimes p', q \otimes q']\text{-}\mathbf{Coalg}$$

derived from the map of polynomials  $[p, q] \otimes [p', q'] \rightarrow [p \otimes p', q \otimes q']$  given on positions by

$$\mathbf{Poly}(p, q) \times \mathbf{Poly}(p', q') \xrightarrow{\otimes} \mathbf{Poly}(p \otimes p', q \otimes q')$$

and on directions by, for  $\phi: p \rightarrow q$  and  $\phi': p' \rightarrow q'$ ,

$$\left( \sum_{I \in p(1)} q[\phi_1(I)] \right) \times \left( \sum_{I' \in p'(1)} q'[\phi'_1(I')] \right) \longleftarrow \sum_{(I, I') \in p(1) \times p'(1)} q[\phi_1(I)] \times q'[\phi'_1(I')]$$

sending  $(I, I', j, j')$  to  $(I, j, I', j')$ .

Concretely, this tensor product takes a  $[p, q]$ -coalgebra  $\mathbf{S}$  and a  $[p', q']$ -coalgebra  $\mathbf{S}'$  to the  $[p \otimes p', q \otimes q']$ -coalgebra with states  $S \times S'$ , action

$$S \times S' \rightarrow \mathbf{Poly}(p, q) \times \mathbf{Poly}(p', q') \rightarrow \mathbf{Poly}(p \otimes p', q \otimes q'),$$

and update described similarly componentwise. The tensor product of coalgebra morphisms is also given by the cartesian product of functions, and it is (very) tedious but ultimately straightforward to check that the essential uniqueness of products guarantees that  $\otimes$  gives a monoidal structure on  $\mathbb{O}\mathbf{rg}$ .

### 2.4 $\mathbb{O}\mathbf{rg}$ as a double category

Defining  $\mathbb{O}\mathbf{rg}$  as a monoidal bicategory is sufficient for most of the constructions of  $\mathbb{O}\mathbf{rg}$ -enriched structures in Section 3. However, using a double category structure casting singleton coalgebras  $\mathbf{S}_\phi \in [p, q]\text{-}\mathbf{Coalg}$  (see Example 2.3) as morphisms  $\phi: p \rightarrow q$  in  $\mathbf{Poly}$  facilitates our eventual definition of maps between dynamic structures.

Specifically, the definition of  $\mathbb{O}\mathbf{rg}$  as a monoidal bicategory extends to a monoidal (pseudo-)double category with coalgebras as horizontal morphisms, maps in  $\mathbf{Poly}$  as vertical morphisms, and squares as in (4) given by maps of coalgebras from  $\mathbf{S} \circ \{\psi\}$  to  $\{\phi\} \circ \mathbf{S}'$ .

$$\begin{array}{ccc} p & \xrightarrow{\mathbf{S}} & q \\ \phi \downarrow & \Downarrow & \downarrow \psi \\ p' & \xrightarrow{\mathbf{S}'} & q' \end{array} \quad (4)$$

The symbol  $\Downarrow$  is intended to indicate that the map is “dynamic”, changing in response to what flows between  $p$  and  $q$ .

As  $\{\phi\}$  and  $\{\psi\}$  have only one state, and composition of coalgebras acts as the cartesian product on states, such a square amounts to a function  $S \rightarrow S'$  making (5) commute:

$$\begin{array}{ccccc} S & \xrightarrow{\beta} & [p, q] \triangleleft S & \xrightarrow{\psi_*} & [p, q'] \triangleleft S \\ f \downarrow & & & & \downarrow [p, q'] \triangleleft f \\ S' & \xrightarrow{\beta'} & [p', q'] \triangleleft S' & \xrightarrow{\phi^*} & [p, q'] \triangleleft S' \end{array} \quad (5)$$

Identities and composites for these squares are determined by the bicategory structure, as this double category is a restriction in the vertical direction of the double category of lax-commuting squares in a bicategory.<sup>3</sup>

We now proceed to discuss various categorical structures enriched in  $\mathbb{O}\mathbf{rg}$ , which describe dynamical systems equipped with extra algebraic structure that allows us to remove abstraction barriers when considering nested layers and complex arrangements of the components of the system.

### 3 Dynamic structures via $\mathbb{O}\mathbf{rg}$ -Enrichment

A monoidal double category is a viable setting for enriching various categorical structures. Intuitively, enrichment in  $\mathbb{O}\mathbf{rg}$  replaces the usual set of arrows between two objects in a categorical structure with a  $[p, q]$ -coalgebra for some choice of polynomials  $p, q$ . Therefore not only can each arrow be realized as a map of polynomials  $p \rightarrow q$ , but this map carries dynamics that encode how a position in  $p$  and a direction in  $q$  determine a transition from one arrow to another. The morphism “reacts” to what’s flowing between  $p$  and  $q$ .

Different situations call for different categorical structures to model their dynamics: some systems primarily involve many-to-one arrangements such as the wiring diagrams in Fig. 1, others such as gradient descent fit naturally into a many-to-many arrow framework, and we expect in future work to consider evolving systems in which different components operate at differing time scales. Rather than choose one such categorical form to favor, and then go through the tedious exercise of forcing all of the others to conform to it, we describe how to add dynamics to the definitions of many different structures.

*A dynamic \*thing\* is a \*thing\* enriched in  $\mathbb{O}\mathbf{rg}$ .*

<sup>3</sup>It should be noted however that the vertical arrows in  $\mathbb{O}\mathbf{rg}$  are regarded as polynomial maps rather than coalgebras, so that they compose strictly unittally and associatively.



This slogan is intentionally imprecise, so as to be maximally inclusive of different notions of categorical structures (\*things\*) and notions of enrichment, and also to allow the reader who has an intuitive understanding and no need for precision to skip the remainder of this paragraph. Our intuition and examples come from the theories of enrichment described in [Lei99] and [Sha22]. In the former, a \*thing\* can be any suitable type of generalized multicategory, while in the latter a \*thing\* can be any structure defined as an algebra for a familial monad on a presheaf category equipped with a choice of “higher” and “lower” dimensional cell shapes. In both cases, \*things\* are algebras for a particular cartesian monad  $T$  which admit an “enriched” analogue with respect to any  $T$ -multicategory. To define  $T$ -algebras enriched in  $\mathbf{Org}$  is then to identify  $\mathbf{Org}$  with a  $T$ -multicategory, and in all of our examples this identification arises naturally from the observation that monoidal double categories give rise to  $T$ -multicategories in a natural way.

We give specific instances of  $\mathbf{Org}$ -enrichment in the following sections: in Section 3.1 for dynamical categories, in Section 3.2 for dynamical multicategories and operads, and in Section 3.3 for dynamical monoidal categories and PRO(P)s. We are also interested in using dynamic duoidal categories to describe dynamical systems in which different contributors to a system operate at different rates, using the duoidal structure on  $\mathbf{Org}$  based on  $\triangleleft$ , but that is beyond the scope of this paper.

### 3.1 Dynamic categories

Enrichment of categories only uses  $\mathbf{Org}$ ’s double category structure—not its monoidal structure—as any double category forms an  $fc$ -multicategory (also known as a virtual double category) in the sense of [Lei99, Section 2.1]. The following definition of enrichment in  $\mathbf{Org}$  is an unwinded version the more general definition in [Lei99, Section 2.2].

**Definition 3.1.** An  $\mathbf{Org}$ -enriched (henceforth *dynamic*) category  $A$  consists of

- a set  $A_0$  of objects;
- for each  $a \in A_0$ , a polynomial  $p_a$ ;
- for each  $a, b \in A_0$ , a  $[p_a, p_b]$ -coalgebra  $\mathbf{S}_{a,b}$ ;
- for each  $a \in A_0$ , an “identitor” square in  $\mathbf{Org}$  as in (6) left; and
- for each  $a, b, c \in A_0$ , a “compositor” square in  $\mathbf{Org}$  as in (6) right:

$$\begin{array}{ccc}
 p_a & \xrightarrow{\{\text{id}_{p_a}\}} & p_a \\
 \parallel & \Downarrow & \parallel \\
 p_a & \xrightarrow{\mathbf{S}_{a,a}} & p_a
 \end{array}
 \qquad
 \begin{array}{ccccc}
 p_a & \xrightarrow{\mathbf{S}_{a,b}} & p_b & \xrightarrow{\mathbf{S}_{b,c}} & p_c \\
 \parallel & & \Downarrow & & \parallel \\
 p_a & \xrightarrow{\mathbf{S}_{a,c}} & & & p_c
 \end{array}
 \tag{6}$$

such that these squares satisfy unit and associativity equations.  $\diamond$

The sets  $\mathbf{S}_{a,b}$  form an ordinary category which we say *underlies*  $A$ . In fact, a dynamic category could be equivalently defined as an ordinary category such that each object  $a$  is assigned a polynomial  $p_a$  and each set of arrows  $\text{Hom}(a, b)$  is the set of states for an assigned  $[p_a, p_b]$ -coalgebra  $\mathbf{S}_{a,b}$ , with composition and identities respecting the coalgebra structure. This means that the arrow  $\text{id}_a$  in  $\text{Hom}(a, a)$  acts as the identity map on  $p_a$  and is unchanged by updates, while for  $f$  in  $\text{Hom}(a, b)$  and  $g$  in  $\text{Hom}(b, c)$  the composite  $f \circ g$  acts



as the composite  $p_a \rightarrow p_b \rightarrow p_c$  of the actions of  $f$  and  $g$ , and the update of their composite equals the composite of their updates.

## 3.2 Dynamic multicategories

A monoidal double category also gives rise to an *fm*-multicategory in the sense of [Lei99, Section 3.1], so we can talk about multicategories enriched in  $\mathbb{O}rg$  as in [Lei99, Section 3.2].

**Definition 3.2.** An  $\mathbb{O}rg$ -enriched (henceforth *dynamic*) multicategory  $A$  consists of

- a set  $A_0$  of objects;
- for each  $a \in A_0$ , a polynomial  $p_a$ ;
- for each  $a_1, \dots, a_n, b \in A_0$ , a  $[p_{a_1} \otimes \dots \otimes p_{a_n}, p_b]$ -coalgebra  $\mathbf{S}_{a_1, \dots, a_n; b}$ ;
- for each  $a \in A_0$ , an “identitor” square in  $\mathbb{O}rg$  as in (7) left; and
- for each  $a_{1,1}, \dots, a_{1,m_1}, \dots, a_{n,1}, \dots, a_{n,m_n}, b_1, \dots, b_n$ , and  $c \in A_0$ , a “compositor” square in  $\mathbb{O}rg$  as in (7) right

$$\begin{array}{ccccc}
 p_a & \xrightarrow{\{\text{id}_{p_a}\}} & p_a & & p_{a_{1,1}} \otimes \dots \otimes p_{a_{n,m_n}} \xrightarrow{\bigotimes_i \mathbf{S}_{a_{i,1}, \dots, a_{i,m_i}; b_i}} p_{b_1} \otimes \dots \otimes p_{b_n} \xrightarrow{\mathbf{S}_{b_1, \dots, b_n; c}} p_c \\
 \parallel & \Downarrow & \parallel & & \parallel \\
 p_a & \xrightarrow{\mathbf{S}_{a,a}} & p_a & & p_{a_{1,1}} \otimes \dots \otimes p_{a_{n,m_n}} \xrightarrow{\mathbf{S}_{a_{1,1}, \dots, a_{n,m_n}; c}} p_c
 \end{array} \quad (7)$$

such that these squares satisfy unit and associativity equations (Definition A.1).  $\diamond$

The sets  $\mathbf{S}_{a,b}$  form an ordinary (set-enriched) multicategory, which underlies  $A$  and has a description similar to the underlying category we described below Definition 3.1.

We will mostly be interested in what we call a *dynamic operad*, the case when a dynamic multicategory  $A$  has only one object, assigned the polynomial “interface”  $p$ . It consists simply of a  $[p^{\otimes n}, p]$ -coalgebra  $\mathbf{S}_n$  for each  $n \in \mathbb{N}$ , equipped with coalgebra maps

$$\{\text{id}_p\} \rightarrow \mathbf{S}_1 \quad \text{and} \quad \bigotimes_{i \in I} \mathbf{S}_{n_i} \rightarrow \mathbf{S}_N \quad (8)$$

where  $I$  is any finite set and  $N := \sum_{i \in I} n_i$ , which together satisfy the usual equations.

**Example 3.3.** A *collective* (as defined in [NS21]) is a  $\otimes$ -monoid in  $\mathbf{Poly}$ , meaning a polynomial  $p$  equipped with a monoid structure on its positions  $p(1)$  and co-unital co-associative “distribution” functions  $p[I \cdot J] \rightarrow p[I] \times p[J]$  for each  $I, J \in p(1)$ . This can be viewed as a dynamic operad where  $\mathbf{S}_n$  is given by  $\{\cdot_n\}$ , the singleton coalgebra on the  $n$ -ary monoidal product  $(\cdot_n): p^{\otimes n} \rightarrow p$ , and where the maps of coalgebras in (8) are isomorphisms deduced from the equations for a monoid.  $\diamond$

**Example 3.4.** In Example 3.3, the coalgebras  $\mathbf{S}_n$  are determined by a single map of polynomials, with trivial updates since the state sets are singletons. This can be generalized to an intermediate notion between collectives and dynamic multicategories, where the coalgebras are still static but may have multiple states.

Given any multicategory  $M$  and multifunctor  $F: M \rightarrow \mathbf{Poly}$ , where  $\mathbf{Poly}$  here denotes the multicategory underlying  $(\mathbf{Poly}, y, \otimes)$ , there is a dynamic multicategory  $A_F$  with

- object set  $\text{Ob}(M)$ ;

- for each  $a \in \text{Ob}(M)$ , the polynomial interface  $p_a := F(a)$ ;
- for each tuple  $(a_1, \dots, a_n; b)$  in  $\text{Ob}(M)$ , state set  $S_{a_1, \dots, a_n; b} := M(a_1, \dots, a_n; b)$ ;
- the action  $\text{act}^\beta: M(a_1, \dots, a_n; b) \rightarrow \mathbf{Poly}(p_{a_1} \otimes \dots \otimes p_{a_n}, p_b)$  is given by  $F$ ; and
- this coalgebra is static, in that for any state  $s$  in  $M(a_1, \dots, a_n; b)$ , the update function  $\text{upd}_s^\beta$  is the constant function at  $s$ .  $\diamond$

**Example 3.5.** Let  $\mathbf{S}$  be any  $p$ -coalgebra for a polynomial  $p$ . There is a dynamic operad on  $p$  with  $\mathbf{S}_0 := \mathbf{S}$ , with  $\mathbf{S}_1 := \{\text{id}_p\}$ , and with all other  $\mathbf{S}_n := \emptyset$  assigned the empty coalgebra.  $\diamond$

**Example 3.6.** Consider a dynamic operad with interface  $y \in \mathbf{Poly}$ . The internal hom polynomial  $[y^{\otimes n}, y]$  is simply  $y$ , so this structure amounts to an operad  $\mathcal{S}$  with a function  $S_n \rightarrow S_n$  for each  $n$ , commuting with the operad structure. A dynamic operad on  $y$  can thus be identified with an operad  $\mathcal{S}$  equipped with an operad map  $\mathcal{S} \rightarrow \mathcal{S}$  to itself.  $\diamond$

### 3.3 Dynamic monoidal categories

A monoidal double category is precisely a representable *fmc*-multicategory as in [Sha22, Section 2], so we can also enrich strict monoidal categories in  $\mathbf{Org}$ .<sup>4</sup> These are similar to  $\mathbf{Org}$ -enriched multicategories, but include many-to-many coalgebras rather than just many-to-one.

**Definition 3.7.** An  $\mathbf{Org}$ -enriched (henceforth *dynamic*) strict monoidal category  $A$  consists of

- a monoid  $(A_0, e, *)$  of objects;
- for each  $a \in A_0$ , a polynomial  $p_a$ ;
- an isomorphism of polynomials  $y \cong p_e$ ;
- for each  $a, a' \in A_0$ , an isomorphism of polynomials  $p_a \otimes p_{a'} \cong p_{a*a'}$ ;
- for each  $a, b \in A_0$ , a  $[p_a, p_b]$ -coalgebra  $\mathbf{S}_{a,b}$ ;
- for each  $a \in A_0$ , an “identitor” square in  $\mathbf{Org}$  as in Eq. (9) left;
- for each  $a, b, c \in A_0$ , a “compositor” square in  $\mathbf{Org}$  as in Eq. (9) center; and
- for each  $a, a', b, b' \in A_0$ , a “productor” square in  $\mathbf{Org}$  as in Eq. (9) right:

$$\begin{array}{ccccc}
 p_a \xrightarrow{\{\text{id}_{p_a}\}} p_a & p_a \xrightarrow{\mathbf{S}_{a,b}} p_b \xrightarrow{\mathbf{S}_{b,c}} p_c & p_a \otimes p_{a'} \xrightarrow{\mathbf{S}_{a,b} \otimes \mathbf{S}_{a',b'}} p_b \otimes p_{b'} & & \\
 \parallel \quad \downarrow \quad \parallel & \parallel \quad \downarrow \quad \parallel & \wr \parallel \quad \downarrow \quad \wr \parallel & & (9) \\
 p_a \xrightarrow{\mathbf{S}_{a,a}} p_a & p_a \xrightarrow{\mathbf{S}_{a,c}} p_c & p_{aa'} \xrightarrow{\mathbf{S}_{a*a', b*b'}} p_{b*b'} & & 
 \end{array}$$

satisfying unit, associativity, and interchange equations.  $\diamond$

Similar to Sections 3.1 and 3.2, the sets  $S_{a,b}$  form the arrows in an ordinary strict monoidal category underlying  $A$ .

For the rest of this paper, we will only be interested in the restricted case of a dynamic monoidal category with object monoid  $(\mathbb{N}, 0, +)$ , which we call a dynamic PRO.<sup>5</sup> Concretely, this consists of a polynomial interface  $p$  (so that in the notation above  $p_n := p^{\otimes n}$  for  $n \in \mathbb{N}$ )

<sup>4</sup>We use throughout the notion *strong* enrichment in a monoidal double category from [Sha22, Section 3].

<sup>5</sup>A PRO is the non-symmetric version of a PROP. While all of our examples are in fact symmetric, to keep the paper short we do not describe their symmetry operations.

along with a  $[p^{\otimes m}, p^{\otimes n}]$ -coalgebra  $S_{m,n}$  for each  $m, n \in \mathbb{N}$ , equipped with the maps of coalgebras as in (9). The identitors, compositors, productors, and their equations amount to the ability to compose any string diagram of the usual sort for monoidal categories, with each  $m$ -to- $n$  box given by a state in  $S_{m,n}$ , into a new box (i.e. state) with the appropriate sources and targets. We denote a dynamic PRO as  $(p, \mathbf{S})$ , where  $\mathbf{S}$  encodes all of the coalgebras  $S_{m,n}$  that constitute the  $\mathbf{Org}$ -enrichment and the structure maps are implicit.

We now turn to morphisms between dynamic PROs; the interested reader can hopefully find analogous definitions for dynamic categories and operads.

**Definition 3.8.** A *morphism* of dynamic PROs from  $(p, \mathbf{S})$  to  $(p', \mathbf{S}')$  is given by a map of polynomials  $\phi: p \rightarrow p'$  and, for each  $m, n \in \mathbb{N}$ , “commutor” squares as in (10) in  $\mathbf{Org}$  which commute with the identitor, compositor, and productor squares.

$$\begin{array}{ccc}
 p^{\otimes m} & \xrightarrow{S_{m,n}} & p^{\otimes n} \\
 \phi^{\otimes m} \downarrow & \Downarrow & \downarrow \phi^{\otimes n} \\
 p'^{\otimes m} & \xrightarrow{S'_{m,n}} & p'^{\otimes n}
 \end{array} \tag{10}$$

◇

This definition of morphism (taken from [Sha22, Section 3]) is the direct theoretical benefit of treating  $\mathbf{Org}$  as a monoidal double category rather than as a monoidal bicategory (closer to its description in [Spi21]). Otherwise morphisms could either only be easily defined between dynamic PROs with the same interface polynomial, which is needlessly restrictive, or take the form of a  $[p, p']$ -coalgebra, which seems to be too general to be of much use.

A morphism  $(p, \mathbf{S}) \rightarrow (p', \mathbf{S}')$  can be interpreted as a way of telling the codomain how to run the domain. The map of polynomials  $p \rightarrow p'$  specifies how the positions of  $p$  can be modeled by those of  $p'$  and how the directions of  $p'$  are returned as directions of  $p$ , while the commutor squares describe how the states of  $\mathbf{S}_{m,n}$  can be modeled by those of  $\mathbf{S}'_{m,n}$  in a way that respects this change of interface. A type of theorem that we hope to instantiate in future work is of the form “this dynamic structure that we’re interested in can be run by (has a map to) this other dynamic structure that we already understand well.”

**Example 3.9.** For a fixed polynomial  $p$ , there is a terminal dynamic PRO with interface  $p$ , which we denote  $\mathbf{S}^{p!}$ ; here  $\mathbf{S}^{p!}_{m,n}$  is the terminal  $[p^{\otimes m}, p^{\otimes n}]$ -coalgebra for each  $m, n \in \mathbb{N}$ .

A state in  $\mathbf{S}^{p!}$  is a (not necessarily finite)  $[p^{\otimes m}, p^{\otimes n}]$ -tree. By this we mean a tree co-inductively defined by a root node labeled with a polynomial map  $\phi: p^{\otimes m} \rightarrow p^{\otimes n}$  together with an arrow—whose source is the root and whose target is another  $[p^{\otimes m}, p^{\otimes n}]$ -tree—assigned to each tuple

$$((I_1, \dots, I_m), i_1, \dots, i_n) \in p^{\otimes m}(1) \times p^{\otimes n}[\phi(I_1, \dots, I_m)] \tag{11}$$

The action of such a tree is simply the map  $\phi$  labeling its root, and the update sends a tuple as in (11) to the target of its assigned arrow.

The idea is that the state-set of the terminal dynamic PRO encodes all possible trajectories along different actions, and this coalgebra is terminal because from any other coalgebra

there is a map to  $\mathbf{S}_{m,n}^{p!}$  sending each state to the tree whose root is labeled by the action of the state and whose edges from the root go to the trees for each of the state's possible updates.

To define a dynamic PRO structure on the terminal coalgebra  $\mathbf{S}^{p!}$ , it only remains to define maps of coalgebras as in Eq. (9), and these are all taken to be the unique map to the terminal  $[p^{\otimes m}, p^{\otimes n}]$ -coalgebra; the equations hold automatically. This is the terminal dynamic PRO with interface  $p$  because for any other such dynamic PRO there is a morphism given by the identity map on  $p$  and with commutor squares to  $\mathbf{S}_{m,n}^{p!}$  the unique map to the terminal  $[p^{\otimes m}, p^{\otimes n}]$ -coalgebra. In other words,  $\mathbf{S}^{p!}$  *uniquely runs* any other dynamic PRO with interface  $p$ .  $\diamond$

## 4 Dynamic Structures in Nature

Our main results are that dynamic structures describe phenomena we see instantiated around us. In this paper, we focus on deep learning and a prediction market in which the reputations of various guess-makers evolve based on how successful they are.

### 4.1 The prediction market dynamic operad

Fix a finite set  $X$ , elements of which we call *outcomes* and intuit to be “all equally likely”, define the set  $\Delta_X^+$  of *guesses on  $X$*  as<sup>6</sup>

$$\Delta_X^+ := \left\{ \gamma: X \rightarrow (0, 1] \mid 1 = \sum_x \gamma(x) \right\}$$

The monoid  $((0, 1], 1, *)$  of nonzero subunitary reals acts on  $\Delta_X^+$  by scalar multiplication, i.e. for any  $0 < m \leq 1$  and  $\gamma \in \Delta_X^+$ , we can define  $m \cdot \gamma \in \Delta_X^+$  as follows:

$$m \cdot \gamma := (x \mapsto m\gamma(x))$$

Let  $\Delta^+$  denote the operad of finite nowhere-zero probability distributions, where  $\Delta_N^+$  is defined as above with the natural number  $N$  regarded as the  $N$ -element set. Then  $\Delta_X^+$  is an algebra for it: for any  $\mu \in \Delta_N$  and  $\gamma \in (\Delta_X^+)^N$ , we define

$$\mu \cdot \gamma := \left( x \mapsto \sum_{i \in N} (\mu_i \cdot \gamma_i)(x) \right)$$

and it is easy to check that  $(\mu \cdot \gamma) \in \Delta_X^+$ , i.e. its components are in bounds  $(\mu \cdot \gamma)(x) \in (0, 1]$  and it is normalized  $\sum_x (\mu \cdot \gamma)(x) = 1$ .

We now construct a dynamic operad with interface  $p_X \in \mathbf{Poly}$  defined as:

$$p_X := \Delta_X^+ y^X$$

---

<sup>6</sup>The assumption that every possible outcome is given some nonzero probability in each guess could be interpreted either as humility of the guess-makers or a strategic decision to avoid permanent loss of all trust. But the real reason for the choice is that it lets us avoid dividing by zero when updating.

and use the  $\Delta_N^+$  as our state spaces. The idea is that a state  $\mu \in \Delta_N^+$  says how much the organization trusts each of its  $N$  members (guess-makers) relative to each other. A member's position at a given moment is a report of how much confidence it has in each of the  $X$ -many possibilities, represented by its probability distribution.

The action of a trust distribution  $\mu \in \Delta_N^+$  is the map of polynomials  $p_X^{\otimes N} \rightarrow p_X$  which on positions sends  $\gamma \in (\Delta_X^+)^N$  to  $\mu \cdot \gamma$  and on directions sends  $x \in X$  to  $(x, \dots, x) \in X^N$ . The idea is that the organization aggregates its members' predictions according to its current trust-distribution, and the outcome is accurately communicated back to each member.

The most interesting part of the dynamic structure is how the trust distribution is updated once predictions are made and a result  $x \in X$  is returned. When  $N = 0$ , there's nothing to do:  $\Delta_0^+ = \emptyset$ . For membership  $N \geq 1$ , trust distribution  $\mu \in \Delta_N^+$ , guesses  $\gamma \in (\Delta_X^+)^N$ , and outcome  $x \in X$ , we define the updated trust distribution  $\gamma(x) * \mu \in \Delta_N^+$  as

$$\gamma(x) * \mu := \left( i \mapsto \frac{\gamma_i(x)\mu_i}{\sum_j \gamma_j(x)\mu_j} \right).$$

Finally, we describe the operadic structure maps. As  $\Delta_1^+$  is a singleton set whose action is the identity on  $p_X$ , the identitor  $\{\text{id}_{p_X}\} \rightarrow \Delta_1^+$  is an isomorphism. The operadic compositor is given by the usual operad structure on (nowhere-zero) distributions:

$$\Delta_N^+ \times \Delta_{M_1}^+ \times \dots \times \Delta_{M_N}^+ \rightarrow \Delta_{\sum_i M_i}^+ \quad (\mu, v_1, \dots, v_N) \mapsto \mu \circ v := ((i, j) \mapsto \mu_i v_j).$$

**Theorem 4.1.** *The maps defined above are maps of coalgebras and satisfy the coherence equations of a dynamic operad described in Definition A.1.*

This is proven in Appendix B.

## 4.2 The gradient descent dynamic PRO

Deep learning uses the algorithm of gradient descent to optimize a choice of function, based on external feedback on its output. This naturally fits into the paradigm of dynamic structures, since functions  $\mathbb{R}^m \rightarrow \mathbb{R}^n$  can form the states of a polynomial coalgebra, with the feedback providing the information needed to update the choice of function. These functions can be composed and juxtaposed in a way that preserves the updates. That is, the composite of gradient descenders is a gradient descender.

**Definition 4.2.** For the rest of this section, we will use the state sets

$$S_{m,n} := \{(M \in \mathbb{N}, f: \mathbb{R}^{M+m}, p \in \mathbb{R}^M) \mid f \text{ is differentiable}\}. \quad \diamond$$

The idea is that these states are the possible parameters among which a gradient descender is meant to find the optimal choice, while  $f$  dictates how the parameter affects the resulting function  $f(p, -)$ . In the dynamics of these states described below, only the value of the parameter  $p$  will be updated; the parameter-space dimension  $M$  and the parameterized function  $f$  will remain fixed, though network composition of gradient descenders will involve combining these data in nontrivial ways. Fix  $\epsilon > 0$ .

For every  $x \in \mathbb{R}$ , let  $T_x\mathbb{R}$  denote the tangent space at  $X$ ; for all practical purposes  $T_x\mathbb{R}$  can be regarded as simply  $\mathbb{R}$ , but in both the description of polynomials as bundles and the intuition for this example it makes sense to use the tangent space at  $x$ . We proceed to define a dynamic PRO with interface  $t := \sum_{x \in \mathbb{R}} y^{T_x\mathbb{R}}$  and with coalgebras  $\mathbf{S}_{m,n}$  which update the state sets  $S_{m,n}$  from Definition 4.2 using gradient descent. The PRO structure maps encode how networks of gradient descenders can be composed into a single gradient descender with a larger parameter space.

**Definition 4.3.** The  $[t^{\otimes m}, t^{\otimes n}]$ -coalgebra structure on  $S_{m,n}$  is given by

- On positions, the action  $\text{act}_{M,f,p}^\beta: \mathbb{R}^m \rightarrow \mathbb{R}^n$  is given by  $f(p, -)$ .
- For  $x \in \mathbb{R}^m$ , the action  $\text{act}_{M,f,p}^\beta(x, -): T_{f(p,x)}\mathbb{R}^n \rightarrow T_x\mathbb{R}^m$  on directions sends  $y \in T_{f(p,x)}\mathbb{R}^n$  to  $\pi_m(Df)^\top \cdot y$ .
- The update function  $\text{upd}_{M,f,p}^\beta$  sends  $x \in \mathbb{R}^m$  and  $y \in T_{f(p,x)}\mathbb{R}^n$  to  $(M, f, p + \epsilon \pi_M(Df)^\top \cdot y)$  for our fixed  $\epsilon$ .  $\diamond$

The action of a state as a map  $t^{\otimes m} \rightarrow t^{\otimes n}$  is given by applying the parameterized function  $f$  with the parameter  $p$ , resulting in a function  $\mathbb{R}^m \rightarrow \mathbb{R}^n$  as desired. The transpose  $(Df)^\top$  of the derivative of  $f$  sends a feedback vector  $y \in T_{f(p,x)}\mathbb{R}^n$ , which can be interpreted as the difference in  $\mathbb{R}^n$  between the “correct” result for  $x$  and the current approximation  $f(p, x)$ , to the corresponding “correction” to  $(p, x)$  in  $\mathbb{R}^{M+m}$ . The projection of this correction to  $T_x\mathbb{R}^m$  provides the action of the state on directions, which in a network will then be further propagated back to the gradient descender which had output  $x$ . The projection to  $T_p\mathbb{R}^M$  provides the direction and magnitude with which to update the parameters (scaled by the “learning rate”  $\epsilon$ ).

Thus far, we have provided the data of the polynomial  $t$  and the  $[t^{\otimes m}, t^{\otimes n}]$ -coalgebras  $\mathbf{S}_{m,n}$  needed to define a dynamic PRO. We now define the identitor, compositor, and productor morphisms of coalgebras presented by the squares in Definition 3.7.

- The identitors  $\{\text{id}_{t^{\otimes n}}\} \rightarrow \mathbf{S}_{n,n}$  send the unique state in the domain to the state

$$(0, \text{id}_{\mathbb{R}^n}, 0) \in S_{n,n}.$$

- The compositors  $\mathbf{S}_{\ell,m} \circ \mathbf{S}_{m,n} \rightarrow \mathbf{S}_{\ell,n}$  send the pair  $((L, f, p), (M, g, q))$  to

$$\left( M + L, g(-, f(-, -)): \mathbb{R}^{M+L+\ell} \xrightarrow{\text{id} \times f} \mathbb{R}^{M+m} \xrightarrow{g} \mathbb{R}^n, (q, p) \in \mathbb{R}^{M+L} \right).$$

- The productors  $\mathbf{S}_{m,n} \otimes \mathbf{S}_{m',n'} \rightarrow \mathbf{S}_{m+m',n+n'}$  send the pair  $((M, f, p), (M', f', p'))$  to

$$(M + M', (f, f'), (p, p')).$$

These structure maps ensure that whenever two gradient descenders are combined in series or parallel, the resulting composite descender retains the parameter spaces of both. Likewise when the input or output of a descender is wired past some other descender in a network, it does not contribute any new parameters and merely preserves its input/output until plugged into a descender. The following is proven in Appendix B.

**Theorem 4.4.** *The maps defined above are maps of coalgebras and satisfy the coherence equations of a dynamic PRO described in Definition A.2.*

## A Coherence Equations

We now present the equations that must be satisfied by the structure maps in dynamic operads and PROs. While we only provide these for the single-object variant of dynamic multicategories and monoidal categories, respectively, the equations in the general case are entirely analogous.

**Definition A.1.** The equations between the identitors and compositors in a dynamic operad are as follows:

- The left and right unit laws

$$\begin{array}{ccccc}
 p^{\otimes n} & \xrightarrow[\simeq]{\{\text{id}_p\}^{\otimes n}} & p^{\otimes n} & \xrightarrow[\simeq]{s_n} & p \\
 \parallel & \Downarrow & \parallel & \parallel & \parallel \\
 p^{\otimes n} & \xrightarrow[\simeq]{s_1^{\otimes n}} & p^{\otimes n} & \xrightarrow[\simeq]{s_n} & p \\
 \parallel & \Downarrow & \parallel & \parallel & \parallel \\
 p^{\otimes n} & \xrightarrow[\simeq]{s_n} & p & & p
 \end{array} = \begin{array}{ccc}
 p^{\otimes n} & \xrightarrow[\simeq]{s_n} & p \\
 \parallel & \parallel & \parallel \\
 p^{\otimes n} & \xrightarrow[\simeq]{s_n} & p
 \end{array} = \begin{array}{ccccc}
 p^{\otimes n} & \xrightarrow[\simeq]{s_n} & p & \xrightarrow[\simeq]{\{\text{id}_p\}} & p \\
 \parallel & \parallel & \parallel & \Downarrow & \parallel \\
 p^{\otimes n} & \xrightarrow[\simeq]{s_n} & p & \xrightarrow[\simeq]{s_1} & p \\
 \parallel & \parallel & \parallel & \Downarrow & \parallel \\
 p^{\otimes n} & \xrightarrow[\simeq]{s_n} & p & & p
 \end{array} \quad (12)$$

- The associativity law

$$\begin{array}{ccccc}
 p^{\otimes \ell_{1,1}} \otimes \dots \otimes p^{\otimes \ell_{n,m_n}} & \xrightarrow[\simeq]{\bigotimes_{i,j} s_{\ell_{i,j}}} & p^{\otimes m_1} \otimes \dots \otimes p^{\otimes m_n} & \xrightarrow[\simeq]{\bigotimes_i s_{m_i}} & p^{\otimes n} \xrightarrow[\simeq]{s_n} p \\
 \wr \parallel & & \Downarrow & \parallel & \parallel \\
 p^{\otimes (\sum_j \ell_{1,j})} \otimes \dots \otimes p^{\otimes (\sum_j \ell_{n,j})} & \xrightarrow{\quad} & \bigotimes_i s_{\sum_j \ell_{i,j}} & \xrightarrow{\quad} & p^{\otimes n} \xrightarrow[\simeq]{s_n} p \\
 \wr \parallel & & \Downarrow & \parallel & \parallel \\
 p^{\otimes (\sum_{i,j} \ell_{i,j})} & \xrightarrow[\simeq]{s_{\sum_{i,j} \ell_{i,j}}} & p & & p
 \end{array} = \begin{array}{ccccc}
 p^{\otimes \ell_{1,1}} \otimes \dots \otimes p^{\otimes \ell_{n,m_n}} & \xrightarrow[\simeq]{\bigotimes_{i,j} s_{\ell_{i,j}}} & p^{\otimes m_1} \otimes \dots \otimes p^{\otimes m_n} & \xrightarrow[\simeq]{\bigotimes_i s_{m_i}} & p^{\otimes n} \xrightarrow[\simeq]{s_n} p \\
 \parallel & \parallel & \wr \parallel & \Downarrow & \parallel \\
 p^{\otimes \ell_{1,1}} \otimes \dots \otimes p^{\otimes \ell_{n,m_n}} & \xrightarrow{\quad} & \bigotimes_{i,j} s_{\ell_{i,j}} & \xrightarrow{\quad} & p^{\otimes (\sum_i m_i)} \xrightarrow[\simeq]{s_{\sum_i m_i}} p \\
 \wr \parallel & & \Downarrow & \parallel & \parallel \\
 p^{\otimes (\sum_{i,j} \ell_{i,j})} & \xrightarrow[\simeq]{s_{\sum_{i,j} \ell_{i,j}}} & p & & p
 \end{array} \quad (13)$$

◇

These equations derive directly from the definition of operads, namely the associativity and unitality of operadic composition. The equations for dynamic PROs below are similarly derived from the definition of monoidal categories, namely that composition and products of arrows are associative and unital (giving the associativity and unitality equations for compositors and productors) and products are functorial (giving the interchange equations).



**Definition A.2.** The equations between the identitors, compositors, and products in a dynamic PRO are as follows:

- The identitor interchange law

$$\begin{array}{ccc}
 p^{\otimes n} \otimes p^{\otimes n'} & \xrightarrow[\simeq]{\{\text{id}_{p^{\otimes n}}\} \otimes \{\text{id}_{p^{\otimes n'}}\}} & p^{\otimes n} \otimes p^{\otimes n'} \\
 \parallel & \Downarrow & \parallel \\
 p^{\otimes n} \otimes p^{\otimes n'} & \xrightarrow{\quad s_{n,n} \otimes s_{n',n'} \quad} & p^{\otimes n} \otimes p^{\otimes n'} \\
 \wr \parallel & \Downarrow & \wr \parallel \\
 p^{\otimes(n+n')} & \xrightarrow[\simeq]{s_{n+n',n+n'}} & p^{\otimes(n+n')}
 \end{array} = \begin{array}{ccc}
 p^{\otimes n} \otimes p^{\otimes n'} & \xrightarrow[\simeq]{\{\text{id}_{p^{\otimes n}}\} \otimes \{\text{id}_{p^{\otimes n'}}\}} & p^{\otimes n} \otimes p^{\otimes n'} \\
 \wr \parallel & \wr \parallel & \wr \parallel \\
 p^{\otimes(n+n')} & \xrightarrow{\quad \{\text{id}_{p^{\otimes(n+n')}}\} \quad} & p^{\otimes(n+n')} \\
 \parallel & \Downarrow & \parallel \\
 p^{\otimes(n+n')} & \xrightarrow[\simeq]{s_{n+n',n+n'}} & p^{\otimes(n+n')}
 \end{array} \quad (14)$$

- The compositor interchange law

$$\begin{array}{ccc}
 p^{\otimes \ell} \otimes p^{\otimes \ell'} & \xrightarrow[\simeq]{s_{\ell,m} \otimes s_{\ell',m'}} & p^{\otimes m} \otimes p^{\otimes m'} & \xrightarrow[\simeq]{s_{m,n} \otimes s_{m',n'}} & p^{\otimes n} \otimes p^{\otimes n'} \\
 \wr \parallel & \Downarrow & \wr \parallel & \Downarrow & \wr \parallel \\
 p^{\otimes(\ell+\ell')} & \xrightarrow[\simeq]{s_{\ell+\ell',m+m'}} & p^{\otimes(m+m')} & \xrightarrow[\simeq]{s_{m+m',n+n'}} & p^{\otimes(n+n')} \\
 \parallel & \Downarrow & \parallel & \Downarrow & \parallel \\
 p^{\otimes(\ell+\ell')} & \xrightarrow[\simeq]{s_{\ell+\ell',n+n'}} & p^{\otimes(n+n')} & & 
 \end{array} = \begin{array}{ccc}
 p^{\otimes \ell} \otimes p^{\otimes \ell'} & \xrightarrow[\simeq]{s_{\ell,m} \otimes s_{\ell',m'}} & p^{\otimes m} \otimes p^{\otimes m'} & \xrightarrow[\simeq]{s_{m,n} \otimes s_{m',n'}} & p^{\otimes n} \otimes p^{\otimes n'} \\
 \parallel & \Downarrow & \parallel & \Downarrow & \parallel \\
 p^{\otimes \ell} \otimes p^{\otimes \ell'} & \xrightarrow{\quad s_{\ell,n} \otimes s_{\ell',n'} \quad} & p^{\otimes n} \otimes p^{\otimes n'} \\
 \wr \parallel & \Downarrow & \wr \parallel \\
 p^{\otimes(\ell+\ell')} & \xrightarrow[\simeq]{s_{\ell+\ell',n+n'}} & p^{\otimes(n+n')}
 \end{array} \quad (15)$$

- The compositor associativity law

$$\begin{array}{ccc}
 p^{\otimes k} & \xrightarrow[\simeq]{s_{k,\ell}} & p^{\otimes \ell} & \xrightarrow[\simeq]{s_{\ell,m}} & p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n} \\
 \parallel & \Downarrow & \parallel & \Downarrow & \parallel & \Downarrow & \parallel \\
 p^{\otimes k} & \xrightarrow[\simeq]{s_{k,m}} & p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n} \\
 \parallel & \Downarrow & \parallel & \Downarrow & \parallel \\
 p^{\otimes k} & \xrightarrow[\simeq]{s_{k,n}} & p^{\otimes n}
 \end{array} = \begin{array}{ccc}
 p^{\otimes k} & \xrightarrow[\simeq]{s_{k,\ell}} & p^{\otimes \ell} & \xrightarrow[\simeq]{s_{\ell,m}} & p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n} \\
 \parallel & \parallel & \parallel & \Downarrow & \parallel & \Downarrow & \parallel \\
 p^{\otimes k} & \xrightarrow[\simeq]{s_{k,\ell}} & p^{\otimes \ell} & \xrightarrow[\simeq]{s_{\ell,n}} & p^{\otimes n} \\
 \parallel & \Downarrow & \parallel & \Downarrow & \parallel \\
 p^{\otimes k} & \xrightarrow[\simeq]{s_{k,n}} & p^{\otimes n}
 \end{array} \quad (16)$$

- The compositor unit laws

$$\begin{array}{ccc}
 p^{\otimes m} & \xrightarrow[\simeq]{\{\text{id}_{p^{\otimes m}}\}} & p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n} \\
 \parallel & \Downarrow & \parallel & \Downarrow & \parallel \\
 p^{\otimes m} & \xrightarrow[\simeq]{s_{m,m}} & p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n} \\
 \parallel & \Downarrow & \parallel & \Downarrow & \parallel \\
 p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n}
 \end{array} = \begin{array}{ccc}
 p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n} \\
 \parallel & \parallel & \parallel \\
 p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n}
 \end{array} = \begin{array}{ccc}
 p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n} & \xrightarrow[\simeq]{\{\text{id}_{p^{\otimes n}}\}} & p^{\otimes n} \\
 \parallel & \parallel & \parallel & \Downarrow & \parallel \\
 p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n} & \xrightarrow[\simeq]{s_{n,n}} & p^{\otimes n} \\
 \parallel & \Downarrow & \parallel & \Downarrow & \parallel \\
 p^{\otimes m} & \xrightarrow[\simeq]{s_{m,n}} & p^{\otimes n}
 \end{array} \quad (17)$$

- The productor associativity law

$$\begin{array}{ccc}
p^{\otimes m} \otimes p^{\otimes m'} \otimes p^{\otimes m''} & \xrightarrow{s_{m,n} \otimes s_{m',n'} \otimes s_{m'',n''}} & p^{\otimes n} \otimes p^{\otimes n'} \otimes p^{\otimes n''} \\
\wr \parallel & \Downarrow & \wr \parallel \\
p^{\otimes(m+m')} \otimes p^{\otimes m''} & \xrightarrow{s_{m+m',n+n'} \otimes s_{m'',n''}} & p^{\otimes(n+n')} \otimes p^{\otimes n''} \\
\wr \parallel & \Downarrow & \wr \parallel \\
p^{\otimes(m+m'+m'')} & \xrightarrow{s_{m+m'+m'',n+n'+n''}} & p^{\otimes(n+n'+n'')}
\end{array} = \begin{array}{ccc}
p^{\otimes m} \otimes p^{\otimes m'} \otimes p^{\otimes m''} & \xrightarrow{s_{m,n} \otimes s_{m',n'} \otimes s_{m'',n''}} & p^{\otimes n} \otimes p^{\otimes n'} \otimes p^{\otimes n''} \\
\wr \parallel & \Downarrow & \wr \parallel \\
p^{\otimes m} \otimes p^{\otimes(m+m'')} & \xrightarrow{s_{m,n} \otimes s_{m'+m'',n'+n''}} & p^{\otimes n} \otimes p^{\otimes(n'+n'')} \\
\wr \parallel & \Downarrow & \wr \parallel \\
p^{\otimes(m+m'+m'')} & \xrightarrow{s_{m+m'+m'',n+n'+n''}} & p^{\otimes(n+n'+n'')}
\end{array} \quad (18)$$

- The productor unit laws

$$\begin{array}{ccc}
p^{\otimes m} & \xrightarrow{s_{m,n}} & p^{\otimes n} \\
\wr \parallel & \wr \parallel & \wr \parallel \\
p^{\otimes 0} \otimes p^{\otimes m} & \xrightarrow{\{\text{id}_{p^{\otimes 0}}\} \otimes s_{m,n}} & p^{\otimes 0} \otimes p^{\otimes n} \\
\parallel & \Downarrow & \parallel \\
p^{\otimes 0} \otimes p^{\otimes m} & \xrightarrow{s_{0,0} \otimes s_{m,n}} & p^{\otimes 0} \otimes p^{\otimes n} \\
\wr \parallel & \Downarrow & \wr \parallel \\
p^{\otimes m} & \xrightarrow{s_{m,n}} & p^{\otimes n}
\end{array} = \begin{array}{ccc}
p^{\otimes m} & \xrightarrow{s_{m,n}} & p^{\otimes n} \\
\parallel & \parallel & \parallel \\
p^{\otimes m} & \xrightarrow{s_{m,n}} & p^{\otimes n}
\end{array} = \begin{array}{ccc}
p^{\otimes m} & \xrightarrow{s_{m,n}} & p^{\otimes n} \\
\wr \parallel & \wr \parallel & \wr \parallel \\
p^{\otimes m} \otimes p^{\otimes 0} & \xrightarrow{s_{m,n} \otimes \{\text{id}_{p^{\otimes 0}}\}} & p^{\otimes n} \otimes p^{\otimes 0} \\
\parallel & \Downarrow & \parallel \\
p^{\otimes m} \otimes p^{\otimes 0} & \xrightarrow{s_{m,n} \otimes s_{0,0}} & p^{\otimes n} \otimes p^{\otimes 0} \\
\wr \parallel & \Downarrow & \wr \parallel \\
p^{\otimes m} & \xrightarrow{s_{m,n}} & p^{\otimes n}
\end{array} \quad (19)$$

◇

## B Proofs of Dynamic Structure

We now proceed to prove that the coalgebras and structure maps defined above for organized predictions and gradient descent form dynamic structures. In each case, it suffices to show that the structure maps on states preserve coalgebra structure, and that the equations in Definition A.1 or Definition A.2, respectively, are satisfied.

*Proof of Theorem 4.1.* The operad equations are all satisfied as  $\Delta^+$  is known to be an operad, and morphisms of coalgebras are entirely determined by a function between the state sets. It then remains only to show that the identitor and compositor as defined in ?? commute with actions and updates. This is clearly true for the identitor as it is an isomorphism, so we focus on the compositor.

For the compositor to commute with actions on positions is the claim that  $\Delta_X^+$  is an algebra for the operad  $\Delta^+$ ; it means that for  $\mu \in \Delta_N^+$ ,  $v_1 \in \Delta_{M_1}^+$ , ...,  $v_N \in \Delta_{M_N}^+$ , and  $\gamma_{i,j} \in \Delta_X^+$  for  $i = 1, \dots, N$  and  $j = 1, \dots, M_i$ , we have

$$\sum_i \mu_i \left( \sum_j v_j \gamma_{i,j} \right) = \sum_{i,j} (\mu_i v_{i,j}) \gamma_{i,j},$$

which is clearly the case.

The compositor commutes with actions on directions because in  $(\mathbf{S}_{M_1} \otimes \cdots \otimes \mathbf{S}_{M_N}) \circ \mathbf{S}_N$  the action of  $(v_1, \dots, v_N, \mu)$  sends an outcome

$$x \in X = p_X[\mu \cdot (v \cdot \gamma)]$$

to

$$(x, \dots, x) \in X^N = p_X^{\otimes N}[v_1 \cdot \gamma_1, \dots, v_N \cdot \gamma_N]$$

and then to

$$(x, \dots, x) \in X^{\sum_i M_i} = p_X^{\otimes \sum_i M_i}[\gamma_{1,1}, \dots, \gamma_{N,M_N}],$$

while in  $\mathbf{S}_{\sum_i M_i}$  the action of  $\mu \circ v$  sends  $x \in X$  to  $(x, \dots, x) \in X^{\sum_i M_i}$  directly.

It then only remains to show that the compositor commutes with updates. Using the shorthand notation  $v = (v_1, \dots, v_N)$  and  $\gamma = (\gamma_1, \dots, \gamma_N) = (\gamma_{1,1}, \dots, \gamma_{N,M_N})$  already employed above, to show that the composite of the updates of  $\mu, v$  agrees with the update of the composite  $\mu \circ v$  amounts to the equation

$$\gamma(x) * (\mu \circ v) = ((v \cdot \gamma)(x) * \mu) \circ (\gamma(x) * v) \quad (20)$$

for any  $x \in X$ . Here  $v \cdot \gamma$  denotes  $(v_1 \cdot \gamma_1, \dots, v_N \cdot \gamma_N)$  and  $\gamma(x) * v$  denotes  $(\gamma_1(x) * v_1, \dots, \gamma_N(x) * v_N)$ . On the  $(i, j)$ -component of these distributions, (20) unwinds to

$$\frac{\gamma_{i,j}(x)(\mu_i v_{i,j})}{\sum_{i',j'} \gamma_{i',j'}(x)(\mu_{i'} v_{i',j'})} = \left( \frac{\sum_{j'} (v_{i,j'} \gamma_{i,j'}(x)) \mu_i}{\sum_{i'} \sum_{j'} (v_{i',j'} \gamma_{i',j'}(x)) \mu_{i'}} \right) \left( \frac{\gamma_{i,j}(x) v_{i,j}}{\sum_{j'} v_{i,j'} \gamma_{i,j'}(x)} \right),$$

which is easily seen to hold by extracting  $\mu_i$  from the first fraction on the right hand side and cancelling the sums over  $j'$ .  $\square$

*Proof of Theorem 4.4.* The unit and associativity equations follow immediately from associativity and unitality of addition, cartesian products, and function composition. The interchange equations follow from the preservation of 0 under addition and identity functions under cartesian products, the analogous interchange property of function composition and cartesian products of functions, and the fact that the compositors and productors act the same way on the parameters and their dimension.

It then remains only to show that the identitors, compositors, and productors are morphisms of coalgebras. This is immediate for the productors, as each component of the action and update functions respects the cartesian products of functions and parameters that define them, so we proceed only for the identitors and compositors.

For the identitors, the state  $(0, \text{id}_{\mathbb{R}^n}, 0)$  acts as the identity function on  $\mathbb{R}^n$  and on directions by the transpose of its derivative, which is also the identity. The updates in the coalgebras  $\mathbf{S}_{n,n}$  only modify the parameter  $p$ , so as the parameter here is 0-dimensional this state is never changed by the update function, as is the case in the coalgebra  $\{\text{id}_{t^{\otimes n}}\}$ . Therefore this function is a map of coalgebras.

The compositors preserve the component of the action on positions as, for states

$$(L \in \mathbb{N}, f: \mathbb{R}^{L+\ell}, p \in \mathbb{R}^L) \quad \text{and} \quad (M \in \mathbb{N}, g: \mathbb{R}^{M+m}, q \in \mathbb{R}^M),$$

we have

$$g(q, -) \circ f(p, -) = g(-, f(-, -))(q, p, -).$$

This may seem like a trivial rewriting, but it illustrates how the compositor was defined in order for the action to be preserved, as on the left we have the composite of the actions on positions as in  $\mathbf{S}_{\ell,m} \circ \mathbf{S}_{m,n}$ , and on the right we apply the compositor and take the action of the resulting state in  $\mathbf{S}_{\ell,n}$ .

To show that the compositor preserves both the action on directions and the update we note that by the chain rule, for  $x \in \mathbb{R}^\ell$  and  $z \in T_{g(q,f(p,x))}$ ,

$$D(g(-, f(-, -)))^\top z = Df^\top \cdot \pi_m(Dg^\top \cdot z) \in T_{(p,x)}\mathbb{R}^{L+\ell}.$$

Applying  $\pi_\ell$  to both sides above shows that the compositor preserves the action on directions, as on the left we have the action on directions after applying the compositor and on the right we have the composition of the actions of  $(L, f, p)$  and  $(M, g, q)$  on directions as in  $\mathbf{S}_{\ell,m} \circ \mathbf{S}_{m,n}$ .

Finally for updates, we observe by the chain rule that the update rule in  $\mathbf{S}_{\ell,n}$  agrees with that in  $\mathbf{S}_{\ell,m} \circ \mathbf{S}_{m,n}$  under the compositor, as either way for  $x, z$  as above the composite state of  $(L, f, p)$  and  $(M, g, q)$  updates to

$$(M + L, g(-, f(-, -)), (p + \epsilon \pi_L(Df^\top \cdot \pi_m(Dg^\top \cdot z)), q + \epsilon \pi_M(Dg^\top \cdot z))).$$

□

## References

- [Lei99] Tom Leinster. “Generalized enrichment for categories and multicategories”. In: (1999). [arXiv:9901139](#). arXiv: 9901139 [math.CT] (cit. on pp. 8, 9).
- [NS21] Nelson Niu and David I. Spivak. “Collectives: Compositional protocols for contributions and returns”. In: (2021). DOI: [10.48550/ARXIV.2112.11518](#). URL: <https://arxiv.org/abs/2112.11518> (cit. on p. 9).
- [Sha22] Brandon Shapiro. “Enrichment of Algebraic Higher Categories”. In: (2022). *In Preparation* (cit. on pp. 8, 10, 11).
- [Spi21] David I. Spivak. “Learners’ languages”. In: *Proceedings of the 4th Annual Conference on Applied Category Theory*. ACT. Cambridge, UK: EPTCS, 2021 (cit. on pp. 2–6, 11).