Operads, Modules, and Differentiation in Machine Learning

Anonymous

Abstract

We identify an exact connection between the one-marked-input modules of Kříž and May and differentiation in machine learning. Freely linearizing the colors and operations of a smooth computation turns the direct sum of its tangent fibers into an operadic module. The square-zero extension is a quotient of the free vector space on the total tangent bundle by fiberwise affine relations. These constructions recover composition of partial derivatives, reverse accumulation, and gradients for tied parameters. A truncated Weil algebra gives a separate construction for mixed derivatives. We also compute derived coinvariants of finite permutation modules by stabilizers, separating real-valued parameter averaging from higher homology in modular coefficients. Worked examples test tied nonlinear parameters and permutation-invariant pooling. For finite stochastic AI components, signed zero-mass perturbations form another operadic module, whose contraction factors bound consistency errors.

1 Introduction

A neural architecture contains choices of layer, topology, parameter-sharing pattern, and symmetry. Each choice affects the maps that training differentiates. An operad describes legal substitutions among such choices. Its algebras interpret the operations, and its modules describe what happens when one input carries a linear perturbation. The question is whether the last statement can be made literal for smooth architectures. Kříž and May work with operads in chain complexes, whose algebra operations are multilinear. A nonlinear smooth map with two inputs is a function on their Cartesian product. Any use of their module formula must account for that change of ambient category.

We give a passage from smooth semantics to the differential graded setting concentrated in degree zero. For each input type X, the algebra color is the free vector space on X, and the module color is the direct sum of tangent fibers over the points of X. This separates the coefficient supplied by an unmarked input from the direction attached to a marked input. It makes the derivative of each operation multilinear in the enlarged vector spaces even if the original operation is nonlinear. The module axioms then follow from the chain rule. The construction traces back to Part I of the monograph, especially its one-marked-input operations and enveloping algebra [1].

The main results and applications are as follows.

1. A smooth algebra over a typed operad induces a genuine module over its real linearization. Theorem 3.1 proves this statement and identifies the fiberwise derivatives.
2. Proposition 3.2 identifies the square-zero extension with the linearization of the tangent algebra modulo fiberwise affine relations. This describes the information retained by the module.
3. Theorem 4.1 derives reverse accumulation on finite computation graphs. Explicit diagonals account for tied parameters. Section 5 treats mixed second derivatives using nilpotents and records their extra data.
4. Sections 6 and 7 give concrete calculations in learning and in symmetry balancing. Section 8.2 gives a module of probability perturbations and uses it to bound discrepancies in stochastic substitution.

This treatment is mathematical. The worked computations check identities on finite examples. Claims about training performance, generalization, and language-model reliability would require separate empirical or statistical evidence.

2 Modules over operadic algebras

Kříž and May develop operads in the symmetric monoidal category of chain complexes over a commutative ring k, with differential of degree -1 in Parts I and II. We write $\mathcal{O}(n)$ for their parameter complex $C(n)$. A symmetric operad has a right Σ_n-action on $\mathcal{O}(n)$, a unit $k \to \mathcal{O}(1)$,

and substitution maps

$$\mathcal{O}(r) \otimes \bigotimes_{i=1}^{r} \mathcal{O}(n_i) \longrightarrow \mathcal{O}\big(\textstyle\sum_i n_i\big).$$

The associativity, unit, and equivariance axioms make substitution independent of how a tree of operations is assembled. Interchanging factors of degrees p and q produces the Koszul sign $(-1)^{pq}$. An $\mathcal{O}$-algebra A has equivariant chain maps $\theta_n : \mathcal{O}(n) \otimes A^{\otimes n} \to A$ compatible with substitution and the unit. These are Definitions 1.1 and 2.1 of Part I of the monograph [1].

The free algebra monad on complexes is

$$T_{\mathcal{O}}(X) = \bigoplus_{n \geq 0} \mathcal{O}(n) \otimes_{k[\Sigma_n]} X^{\otimes n}.$$

Its multiplication substitutes operations, and its algebra axiom expresses the compatibility of θ with that substitution. Part I, Definition 3.3 and Proposition 3.4, identify $\mathcal{O}$-algebras with algebras over this monad. The tensor powers make the structure maps multilinear in their input vectors. A nonlinear smooth primitive therefore calls for a semantic encoding before these structure maps can represent it.

2.1 The marked slot and the enveloping algebra

For a fixed $\mathcal{O}$-algebra A, a module M in the sense of Kříž and May has chain maps

$$\lambda_n : \mathcal{O}(n) \otimes A^{\otimes (n-1)} \otimes M \longrightarrow M, \qquad n \geq 1. \quad (1)$$

Here the last slot is marked. Equivariance under Σ_{n-1} permutes the unmarked slots. To put M in position i, use the permutation moving that position to the last slot, with the corresponding action on $\mathcal{O}(n)$ and the Koszul signs. Write the resulting operation as $\lambda_{n,i}$. Equivariance under the subgroup fixing the mark makes this description independent of the permutation used. When such operations are nested, every subtree outside the path from the mark to the root is evaluated in A. The operations along the marked path act on M. The module axiom identifies this evaluation with applying the composite operation once (Part I, Definition 4.1).

Define the complex of raw marked operations by

$$V_A = \bigoplus_{n \geq 1} \mathcal{O}(n) \otimes_{k[\Sigma_{n-1}]} A^{\otimes (n-1)}.$$

The action maps assemble into a single map $V_A \otimes M \to M$. The universal enveloping algebra $U_{\mathcal{O}}(A)$ is the quotient of the tensor algebra on V_A by the relations for the unit and for operadic substitution. A product of two marked operations represents insertion of the inner operation at the mark of the outer one. Substitutions in unmarked slots can be evaluated in A before the marked operation acts. The maps in (1) give a representation of the tensor algebra, and the module axioms make it factor through the quotient. Conversely, restricting a $U_{\mathcal{O}}(A)$-action to V_A recovers the maps and their axioms. This proves the identification of categories in Part I, Definition 4.9 and Proposition 4.10. The associative and commutative examples give $A \otimes A^{\mathrm{op}}$ and A, respectively (Part I, Examples 4.11).

2.2 Square-zero extensions

Proposition 2.1. *An $\mathcal{O}$-module M over A determines a split $\mathcal{O}$-algebra extension $A \oplus M \to A$ on which every operation with at least two inputs in M is zero. Conversely, a split extension with this vanishing property determines an A-module on its kernel.*

Proof. Define the operations on $A \oplus M$ by multilinearity. With every argument in A, use θ_n and include the result in the A-summand. With one argument in M, use $\lambda_{n,i}$ and include the result in the M-summand. With two or more arguments in M, set the result to zero. The individual actions are chain maps and are equivariant, so the extended operations have both properties. To check substitution, expand the inputs into their summands. With zero marked inputs, the two evaluations agree by the algebra axiom. With one marked input, they agree by the module axiom. With two or more, either one inner subtree has at least two marked inputs and vanishes, or at least two inner subtrees supply M-inputs to the outer operation, which then vanishes. The operadic unit acts as the identity on both summands, and the projection and inclusion are algebra maps.

For an extension $p : E \to A$ split by an algebra map, use the splitting to write $E = A \oplus \ker p$ as a complex. An operation with one input in $\ker p$ lands in $\ker p$ by multilinearity and compatibility with p. The operadic axioms on these operations are the module axioms. The vanishing requirement determines the remaining operations, so the constructions are inverse under the splitting. □

Suppose $s(a) = (a, D(a))$ is a chain map section of $A \oplus M \to A$. For s to preserve the operations is precisely to satisfy the following derivation law:

$$D\theta_n(c; a_1, \dots, a_n) = \sum_{i=1}^{n} \lambda_{n,i}(c; a_1, \dots, m_i, \dots, a_n), \qquad m_i = D(a_i). \quad (2)$$

The empty sum for $n = 0$ makes D vanish on nullary operations. Conceptually, the sum arises from expanding on each leaf of an operation tree and retaining terms with one mark. Induction over grafting gives the derivative of a composite as the sum of the linear operators obtained by propagating each mark along its path to the root. If several leaves use the same parameter, all its occurrences contribute to the resulting linear map of the shared perturbation.

Example 2.2 (Two-sided sensitivity in a tied matrix block)**.** Take $A = M_r(\mathbb{R})$ with its usual multiplication and use the regular bimodule $M = A$. For fixed $U, V \in A$, let $F(W) = UWV + W^2$. In the associative square-zero extension, the second component of $F(W, H)$ is

$$J_F(W)[H] = UHV + WH + HW.$$

Each summand acts by a two-sided context $H \mapsto aHb$, represented by $a \otimes b^{\mathrm{op}}$ in $A \otimes A^{\mathrm{op}}$. The first summand uses one occurrence of W and the square uses two occurrences. For the Frobenius pairing $\langle G, H\rangle = \mathrm{tr}(G^{\mathsf{T}}H)$, cyclicity of the trace gives

$$J_F(W)^*[G] = U^{\mathsf{T}}GV^{\mathsf{T}} + W^{\mathsf{T}}G + GW^{\mathsf{T}}.$$

In a block followed by a scalar loss, G is the incoming matrix cotangent. The adjoint sends it back to the shared matrix parameter and adds the contributions of its occurrences.

2.3 The scope of homological conversion

The paired monad of Part I, Definition 4.3 and Proposition 4.4, assigns (X, Y) the pair $(T_{\mathcal{O}}(X), V_X \otimes Y)$, where V_X is the same marked-slot expression with X in place of A. Its algebras comprise an operadic algebra and a module over it. In Part II, Section 4, the authors use the two-sided monadic bar construction on such data. For a monad T, a right T-functor F, and a T-algebra A, the simplicial degree q is FT^qA. Its first and last faces use the right action and algebra action, respectively. Interior faces use monad multiplication, and degeneracies use the unit. The augmentation of $B_\bullet(T, T, A)$ has a simplicial contraction after forgetting the algebra structure (Part II, Example 4.2).

With a Σ-projective simplicial operad the conversion theorem treats partial simplicial algebras and modules. For an arity wise quasi-isomorphism of dg operads over a characteristic-zero field, the dg comparison applies to genuine algebras and their modules (Part II, Theorems 1.1 and 1.4). Part V constructs derived tensor and Hom with a cell replacement in its E_∞-module setting, notably Sections 5, 6, and 8. These hypotheses govern applications to complexes, such as coherent interfaces with derived composition. The equivalence with modules over $U_{\mathcal{O}}(A)$ by itself supplies no canonical symmetric monoidal tensor on modules over an arbitrary enveloping algebra. Preservation of numerical predictions or loss requires a separate comparison intertwining the chosen evaluations.

3 Smooth semantics and a linear operadic module

3.1 Operations, substitutions, and base points

Fix a set of colors. A color c has a finite-dimensional real vector space X_c as its interpretation. Colors can include product spaces or a zero-dimensional space. Let P be a symmetric colored operad in sets. An operation $p \in P(c_1, \dots, c_n; d)$ is interpreted by a smooth map

$$f_p : X_{c_1} \times \cdots \times X_{c_n} \longrightarrow X_d.$$

For a nullary operation the product is a point. An interpretation is a smooth P-algebra when it respects the units, permutations, and simultaneous substitution. In particular, substituting q_i into input i gives the map

$$f_{p(q_1,\dots,q_n)}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = f_p\big(f_{q_1}(\boldsymbol{x}_1), \dots, f_{q_n}(\boldsymbol{x}_n)\big).$$

The tuples $\boldsymbol{x}_i$ are distinct input blocks of the substituted operation. Reuse of a variable can be represented by a copy map, as in Section 4. An example is the operad of all smooth maps between products of the chosen spaces, with the block substitution above. Another is the free operad on typed primitives such as an activation, an affine map, or a sum.

Write $\mathcal{O} = \mathbb{R}[P]$ for the real linearization with a basis element $[p]$ for every operation. The bijection between the product basis and the tensor basis gives $\mathbb{R}[S \times T] \cong \mathbb{R}[S] \otimes_{\mathbb{R}} \mathbb{R}[T]$. Thus independently extending substitutions by linearity defines an operad in real vector spaces. We regard it as a dg operad in degree zero. Formal addition of operations takes place in $\mathcal{O}$. Even when $f_p + f_q$ is interpreted by some operation r, the basis element $[r]$ and the formal sum $[p] + [q]$ have different actions in general.

For each color let

$$A_c = \mathbb{R}[X_c], \qquad M_c = \bigoplus_{x \in X_c} T_x X_c.$$

Here $[x]$ denotes a basis vector of A_c and ι_x denotes the inclusion of a tangent fiber into M_c. Both use finite

support. These objects capture point and first-order data algebraically. In particular, the free vector space has no prescribed completion or topology.

3.2 The one-marked-input action

For an input tuple $\boldsymbol{x}$, let $D_i f_p(\boldsymbol{x})$ be the derivative with only the i-th input varying. It maps $T_{x_i}X_{c_i}$ to $T_{f_p(\boldsymbol{x})}X_d$.

Theorem 3.1 (Tangent module over the linearization)**.** *The formulas*

$$A(p)([x_1],\dots,[x_n]) = [f_p(\boldsymbol{x})], \tag{3}$$
$$M_i(p)([x_1],\dots,\iota_{x_i}v,\dots,[x_n]) = \iota_{f_p(\boldsymbol{x})} D_i f_p(\boldsymbol{x})v \tag{4}$$

extend to an $\mathcal{O}$-algebra A and an operadic A-module M. For one color the action is of the form in Kříž and May, Part I, Definition 4.1. The colored action satisfies the same one-marked-input axioms with matching input and output colors.

Proof. The first formula defines a multilinear map on the tensor of all input spaces since it specifies the values on pure basis tensors. In the second formula, fix all unmarked basis points. The map into M_d is linear on each $T_{x_i}X_{c_i}$. The direct sum property extends it to M_{c_i}, and multilinearity in the other basis inputs completes the definition. Finally extend linearly in $[p]$.

Operad substitution holds in A because the interpretation respects substitution. For the module equation take a marked input j of q_i, and put $y_a = f_{q_a}(\boldsymbol{x}_a)$. The partial chain rule gives

$$D_{(i,j)} f_{p(q_1,\dots,q_n)}(\boldsymbol{x}) = D_i f_p(\boldsymbol{y})\, D_j f_{q_i}(\boldsymbol{x}_i). \tag{5}$$

All unmarked blocks use A, so they supply the basis elements $[y_a]$. The inner marked action produces a vector in $T_{y_i}X_{c_i}$, and the outer marked action applies the first factor in (5). This agrees with evaluating the marked slot of the substituted operation. Units and permutations respect the same formulas, since the derivative of the identity is the identity and permutation reindexes the summands. The equations extend from the generating tensors to all inputs by multilinearity. □

Equation (5) means that the derivative in a single input of a tree of operations composes along its path to the output. The unmarked branches compute the base points for the derivative along that path. In the one-color case these compositions act through the enveloping algebra $U_{\mathcal{O}}(A)$ of Kříž and May. In the colored case each one-marked map has a specified source and target color, and the relations equate two composites only when their colors match.

Proposition 3.2 (The affine relations in a tangent fiber)**.** *The smooth tangent operations on the spaces TX_c give an $\mathcal{O}$-algebra on $\mathbb{R}[TX]$. There is a canonical surjective algebra map*

$$Q_c : \mathbb{R}[TX_c] \longrightarrow A_c \oplus M_c, \qquad [(x,v)] \longmapsto ([x], \iota_x v),$$

where $A \oplus M$ is the square-zero extension associated to the module. The kernel is the span, for each fiber, of the relations

$$[(x, v+w)] - [(x,v)] - [(x,w)] + [(x,0)],$$
$$[(x,av)] - a[(x,v)] + (a-1)[(x,0)] \quad (a \in \mathbb{R}).$$

Proof. The tangent map respects substitution by the ordinary chain rule. On a tuple of basis vectors $[(x_i, v_i)]$, the composite of the tangent operation and Q_d gives

$$\Big([f_p(\boldsymbol{x})], \iota_{f_p(\boldsymbol{x})} \sum_i D_i f_p(\boldsymbol{x}) v_i\Big).$$

In the square-zero extension, only the term with no marked input and the terms with exactly one marked input survive. Their values give the same pair. This proves that Q is a morphism on basis vectors and hence on all inputs.

Surjectivity follows from $[(x,0)]$ and $[(x,v)] - [(x,0)]$. Each displayed relation maps to zero. In the quotient by their span, for a fixed x the assignment $v \mapsto [(x,v)] - [(x,0)]$ is linear. Combining this assignment with sending 1 to $[(x,0)]$ constructs the inverse from $\mathbb{R} \oplus T_x X_c$ for that fiber. The sum over fibers supplies the claimed kernel. The relations form an operadic ideal because they are the kernel of the algebra morphism Q. □

For composable unary operations $f : X \to Y$ and $g : Y \to Z$, write $h = g \circ f$. The two routes in the following square have the same effect on every generator.

$$\begin{array}{ccc} \mathbb{R}[TX] & \xrightarrow{\mathbb{R}[Th]} & \mathbb{R}[TZ] \\ \downarrow{\scriptstyle Q_X} & & \downarrow{\scriptstyle Q_Z} \\ A_X \oplus M_X & \xrightarrow{-h^{(1)}\rightarrow} & A_Z \oplus M_Z. \end{array}$$

An implementation therefore needs the fibers visited by the program on the given data. Constructing the full free vector space would add no information to derivative evaluation at those points.

4 Reverse accumulation and parameter sharing

A parameterized layer has the form $F : P \times X \to Y$. An independent composition with $G : Q \times Y \to Z$ has

parameter space $Q \times P$ and evaluates to $G(q, F(p, x))$. Associativity follows from associativity of function composition and the canonical reassociation of products. This is the usual parameterized-map construction appearing in categorical learning [2, 3, 4]. The tangent module supplies the one-input linear pieces of the derivative of the interpretation.

4.1 Finite computation graphs

Fix a finite acyclic graph. Each noninput node w has an ordered list of predecessor slots $(v(w,1), \dots, v(w,r_w))$ and a local smooth operation F_w. A predecessor can occur in several slots. Write z_w for the evaluated state and

$$L_{w,i} = D_i F_w(z_{v(w,1)}, \dots, z_{v(w,r_w)}).$$

The output is one node o, with a scalar loss ℓ of z_o. Tuple-valued outputs can be made into one node by a product. If h_v denotes a tangent perturbation at node v, then

$$h_w = \sum_{i=1}^{r_w} L_{w,i} h_{v(w,i)},$$

by splitting the tangent input of F_w into its slots. Each $L_{w,i}$ appears in the action of Theorem 3.1. For a linear map $L : U \to V$, write $L^\vee : V^* \to U^*$ for precomposition with L.

Theorem 4.1 (Reverse accumulation for a graph)**.** *Initialize the output covector with $D\ell(z_o)$ and all other covectors with zero. Process each noninput node w in reverse topological order. For every slot i, add $L_{w,i}^\vee \lambda_w$ to its predecessor's covector. At the input nodes the final covectors are exactly the input partial differentials of the composite loss.*

Proof. Start with the linear pairing $\langle D\ell(z_o), h_o \rangle$. Whenever a noninput variable h_w is eliminated, use its forward tangent equation to write

$$\langle \lambda_w, h_w \rangle = \sum_i \langle L_{w,i}^\vee \lambda_w, h_{v(w,i)} \rangle.$$

Processing in reverse order puts all downstream contributions at w before it is eliminated. If an input occurs repeatedly, the identical h_v appears with a sum of covectors. At the end the original differential has been written as a pairing with all input variations independently. Its coefficients are the desired partial differentials. □

No inner product is required for the reverse sweep. Choosing an inner product identifies each covector with a gradient. Under Euclidean coordinates $L^\vee$ is represented by the transpose. This is reverse-mode automatic differentiation [11], which also underlies the categorical formulations in [2, 3]. Here its linear pieces are obtained as operators in a module over an explicit linearized operad.

The cotangents of this calculation are local to the evaluated fibers. The algebraic dual of M_c is $\prod_{x \in X_c} T_x^* X_c$. A transpose on the whole dual need not preserve finite support. For example, under the unary map $x \mapsto \sin x$, a covector supported at the output 0 acquires an entry at every $n\pi$ when pulled back. Each such entry is multiplied by $\cos(n\pi) = (-1)^n$. A reverse pass on a fixed graph uses only the fibers specified by its forward evaluation.

4.2 Tying parameters along a diagonal

Let $\Phi : P_1 \times \cdots \times P_r \times X \to Y$ be a model with independent parameter slots. A parameterization $\tau : Q \to \prod_i P_i$ gives the tied model $\Phi(\tau(q), x)$. At fixed x put $L(p) = \ell(\Phi(p, x))$. The tied differential is

$$D(L \circ \tau)(q) = (D\tau(q))^\vee \, DL(\tau(q)).$$

If all $P_i = P$ and $\tau(q) = (q, \dots, q)$, it follows that

$$D(L \circ \tau)(q) = \sum_{i=1}^{r} D_i L(q, \dots, q).$$

The summands are covectors on P. The sum arises from evaluating the module on the smooth diagonal $P \to P^r$, viewed as a unary operation whose output color is a product. Its tangent is $v \mapsto (v, \dots, v)$, so dualizing produces addition. It would give a different result to treat repeated mentions as newly independent parameters.

For equipped parameter spaces, the Euclidean gradient used by an optimizer depends on the metric chosen on Q. In particular, taking the pullback of the product metric to the diagonal rescales the repeated block by r and changes the gradient from a sum to an average. The parameter differential already determines the chain rule and is independent of this decision. Section 7 computes the same distinction for permutation orbits.

4.3 Differentiating a symmetry

Let $\sigma : X \to X$ and $\rho : Y \to Y$ be invertible smooth maps with $F \circ \sigma = \rho \circ F$. The chain rule gives

$$DF(\sigma x)\, D\sigma(x) = D\rho(F(x))\, DF(x).$$

For a linear action and an invariant F, take $\rho = \mathrm{id}$. Then $DF(\sigma x)\sigma = DF(x)$. If the action on X is orthogonal and F is scalar, the corresponding gradient satisfies $\nabla F(\sigma x) = \sigma \nabla F(x)$. This gives a concrete check of a

permutation-invariant architecture and of its backward operation. The pooling example below exhibits the check without assuming a universality theorem for invariant networks.

5 Mixed derivatives from nilpotent coefficients

At a given evaluation point a first-order module operation stores a value and the linear derivative slots. Two maps can have identical first-order data at that point and different quadratic terms. Indeed $t \mapsto t^2$ and $t \mapsto 0$ have equal values and first derivatives at 0, with second derivatives 2 and 0. To retain those terms we use an algebra with two nilpotent parameters.

Definition 5.1. A real Weil algebra for this discussion is a finite-dimensional commutative real algebra $W = \mathbb{R} \oplus \mathfrak{m}$, with $\mathfrak{m}$ a nilpotent ideal and quotient map $W \to \mathbb{R}$. Suppose $\mathfrak{m}^{s+1} = 0$. For a smooth map $F : X \to Y$, the extension to W is

$$F_W(x + \xi) = F(x) + \sum_{a=1}^{s} \frac{1}{a!} D^a F(x)[\xi, \dots, \xi].$$

The nilpotent part ξ lies in $X \otimes \mathfrak{m}$. Here each derivative is extended by multiplying its coefficients in W.

Proposition 5.2 (Composition of finite jets)**.** *Real-space products and composition are preserved by $F \mapsto F_W$. This applies to substitutions among the operations of any smooth P-algebra.*

Proof. Choose coordinates and expand the Taylor polynomial at the real point x through order s. The Taylor polynomial of $G \circ F$ is obtained, to this order, by substituting the polynomial of F at x into that of G at $F(x)$ and discarding terms of degree $> s$. In detail, for real η near zero the omitted term for a smooth map is $O(\|\eta\|^{s+1})$; composing with the other Taylor polynomial preserves this order since $F(x + \eta) - F(x) = O(\|\eta\|)$. Thus the two degree-$\leq s$ polynomials have the same coefficients by uniqueness of Taylor coefficients. On substituting any $\xi \in X \otimes \mathfrak{m}$, the discarded monomials vanish. Tuples of Taylor polynomials give the assertion about products. □

Take the hyperdual algebra $W = \mathbb{R}[\epsilon, \delta]/(\epsilon^2, \delta^2)$ [12]. For $u, v, w \in X$, the coefficient of $\epsilon\delta$ in $F_W(x + \epsilon u + \delta v + \epsilon\delta w)$ is

$$DF(x)w + D^2F(x)[u, v].$$

Using both directions gives, for composable F and G,

$$D^2(G \circ F)(x)[u, v] = DG(F(x))\, D^2F(x)[u, v] + D^2G(F(x))[DF(x)u, DF(x)v].$$

The mixed term involves two independently marked directions. A single module slot cannot hold it as a linear map in just one direction, evaluated at the same base point. With a nonlinear reparameterization this is also the origin of the additional term in the Hessian of $L \circ \tau$. For a linear tie τ, the second derivative of τ is zero, but cross terms in D^2L still contribute. For example, for a scalar tie $\tau(t) = (t, t)$,

$$(L \circ \tau)''(t) = L_{11}(t, t) + L_{12}(t, t) + L_{21}(t, t) + L_{22}(t, t).$$

This provides a direct rule for curvature calculations in tied models.

6 Worked examples in learning

6.1 A nonlinear layer with a tied parameter

Consider a two-stage scalar model with input x, offset b, and initially independent parameters α, β. Write

$$z = \alpha x + b, \quad h = \tanh z, \quad y = \beta h,$$
$$L = \tfrac{1}{2}(y - t)^2,$$

where t is a fixed target. Each operation is smooth on its Euclidean input space. The graph uses the multiplication operation twice and inserts the nonpolynomial activation tanh between the two uses. We first take the marked-slot maps of the untied graph and then impose the sharing relation.

Let $\overline{s}$ denote the cotangent of an intermediate scalar s for this loss. The last multiplication has marked-slot maps given by multiplication by h in the β slot and by β in the h slot. Taking adjoints in the standard scalar pairing gives

$$\overline{y} = y - t, \qquad \overline{\beta} = h\,\overline{y}, \qquad \overline{h} = \beta\,\overline{y}.$$

The activation and affine operation then give

$$\overline{z} = (1 - h^2)\overline{h},$$
$$\overline{\alpha} = x\,\overline{z}, \qquad \overline{x} = \alpha\,\overline{z}, \qquad \overline{b} = \overline{z}.$$

Each expression is determined by a one-slot map and its downstream cotangent. In the algebra-module construction above, the underlying slot is based at its particular forward value. For example, the slot map from $T_z\mathbb{R}$ to $T_h\mathbb{R}$ multiplies by $1 - h^2$, with the coefficient evaluated at the given pass through the graph.

Tie the parameters using the diagonal $\Delta(w) = (\alpha, \beta) = (w, w)$. The parameter variation $\dot{w}$ has image $(\dot{w}, \dot{w})$. Pairing this image with $(\overline{\alpha}, \overline{\beta})$ gives

$$\overline{w} = \overline{\alpha} + \overline{\beta} = (y - t)\big(h + wx(1 - h^2)\big).$$

The first term inside the parentheses comes from the direct scaling of h, and the second comes from the use of w before the activation. In a forward calculation the same two paths give

$$\dot{z} = x\dot{w} + w\dot{x} + \dot{b},$$
$$\dot{y} = h\dot{w} + w(1-h^2)\dot{z}.$$

The coefficient of $\dot{w}$ in $\dot{L} = (y-t)\dot{y}$ agrees with the reverse computation. This calculation separates one-hole operadic substitution along a path from the addition that occurs at a copied input.

For an exact evaluation take $w = x = 1$, $t = 0$, and $b = \operatorname{artanh}(1/2) - 1$. Then $h = y = 1/2$ and $L = 1/8$. The two branches contribute $\overline{\alpha} = 3/8$ and $\overline{\beta} = 1/4$, giving

$$(\partial_w L, \partial_b L, \partial_x L) = (5/8,\ 3/8,\ 3/8).$$

Here $\partial_x L$ measures input sensitivity. For a gradient update of the two model parameters alone with step $1/16$, the new values are $w' = 123/128$ and $b' = b - 3/128$. With x fixed at 1, the new preactivation is $z' = z - 1/16$. Indeed the two parameter changes contribute $5/128$ and $3/128$ to the decrease in z. From $\operatorname{artanh}(1/2) > 1/2$ we have $0 < z' < z$. Strict monotonicity of tanh and $0 < w' < 1$ give $0 < y' < y$, hence the loss decreases for this update. The accompanying script computes its new value.

The same point gives a mixed-derivative check. For $y = w\tanh(wx+b)$, denote the activation by σ. The relevant derivatives can be read off from the affine operation and the two uses of w:

$$y_w = \sigma + wx\sigma', \qquad y_x = w^2\sigma',$$
$$y_{wx} = 2w\sigma' + w^2x\sigma'', \quad L_{wx} = y_w y_x + (y-t)y_{wx}.$$

The values $\sigma' = 3/4$ and $\sigma'' = -3/4$ at the displayed point give $y_{wx} = 3/4$ and $L_{wx} = 21/16$. Equivalently this is the coefficient of $\epsilon\delta$ when varying w by ϵ and x by δ in $\mathbb{R}[\epsilon,\delta]/(\epsilon^2,\delta^2)$. It detects the second derivative of the activation in addition to the compositions of first derivatives.

6.2 Permutation-invariant pooling

We use the sum-pooling architecture of Deep Sets [7] for a fixed number of input elements.

Let the latent space be $Z = \mathbb{R}^2$, and put

$$\mu_n(z_1,\dots,z_n) = \sum_{i=1}^{n} z_i, \qquad \mu_0 = 0.$$

Every permutation of the input slots of μ_n gives the same map. Substituting sums of m_j latent vectors in each slot of μ_k gives $\mu_{\sum_j m_j}$. Thus the sum family gives a commutative operad algebra in smooth spaces, including the nullary and unary structure maps. In the real linearization, the algebra on the free vector space generated by the points of Z is the monoid algebra $\mathbb{R}[Z,+]$, since $[z][z'] = [z+z']$ and the unit is $[0]$. In the usual trivialization $T_zZ = Z$, the derivative of addition in either slot sends a vector unchanged to the fiber over the sum. The resulting tangent module of the linearized sum algebra is $\mathbb{R}[Z,+] \otimes_{\mathbb{R}} Z$, with the first factor acted on by multiplication. The identification sends $[z]\otimes v$ to v in the summand indexed by z.

To obtain a Deep Sets model, use the shared smooth feature map $\phi_a(s) = (as, as^2)$, a sum, and the readout $\rho(u,v) = u^2 + v$. Here a is a trainable scalar. The feature and readout maps are smooth operations in the surrounding colored operad. At a fixed size n the composite is

$$F_a(\mathbf{x}) = \rho\left(\sum_i \phi_a(x_i)\right) = (aS_1)^2 + aS_2,$$
$$S_j = \sum_i x_i^j.$$

A permutation of $\mathbf{x}$ permutes the latent vectors in the same way because every slot uses the same ϕ_a. The sum coequalizes the permutation action, so F_a is invariant and descends set-theoretically to the orbit set. The Euclidean gradient is calculated on the ordered input space. For a permutation matrix P, differentiate $F_a(P\mathbf{x}) = F_a(\mathbf{x})$ in $\mathbf{x}$. The chain rule and $P^{\mathsf{T}}P = I$ give

$$\nabla_{\mathbf{x}} F_a(P\mathbf{x}) = P\nabla_{\mathbf{x}} F_a(\mathbf{x}).$$

Differentiation in a instead gives an invariant scalar $\partial_a F_a(P\mathbf{x}) = \partial_a F_a(\mathbf{x})$. These two transformation laws also hold after composing with a smooth scalar loss and fixing its target.

The readout sends the unit output cotangent to $\lambda = (2aS_1, 1)$. Each pooling slot receives λ, since all marked-slot derivatives of the latent sum are identities in the given trivialization. At slot i, the x_i-derivative of the feature map is $(a, 2ax_i)$, and the a-derivative is (x_i, x_i^2). The pairing and the parameter diagonal now give

$$\partial_{x_i} F_a = 2a^2S_1 + 2ax_i,$$
$$\partial_a F_a = \sum_i (2aS_1x_i + x_i^2) = 2aS_1^2 + S_2.$$

The three stages have distinct roles. The symmetry of the sum establishes forward invariance, the cotangent of the sum distributes the readout covector to each feature branch, and the cotangent of parameter copying adds the resulting parameter covectors.

At $a = 2$ and $\mathbf{x} = (1, -2, 3)$, the sums are $S_1 = 2$, $S_2 = 14$. The resulting values are

$$F_a(\mathbf{x}) = 44, \quad \partial_a F_a = 30, \quad \nabla_{\mathbf{x}} F_a = (20, 8, 28).$$

The parameter contributions from the three branches read $9, -12, 33$ before accumulation. The attached verification script computes these values by a reverse pass and checks all six permutations of this input. Its central differences also check the parameter and input gradients of both examples and the tied model's mixed loss derivative. The latter value is also calculated by direct multiplication and Taylor expansion in the four-dimensional algebra with basis $1, \epsilon, \delta, \epsilon\delta$.

7 Tied parameters and derived symmetry quotients

Suppose a finite set S indexes parameter slots and a finite group G acts on S. Examples include relabeling interchangeable branches and applying automorphisms of a fixed graph to its node slots. Write G_s for the stabilizer of s and S/G for the set of orbits. This symmetry gives two constructions on the same representation. Tying restricts parameters to an invariant subspace. Orbit pooling identifies slots by taking coinvariants. Their relationship depends on the coefficient field and, for gradients, on the metric chosen on the tied parameters.

7.1 Balancing and averaging

Let k be a field, $A = k[G]$, and let M be a left A-module. Regard A as a differential graded algebra concentrated in degree zero, with augmentation $\epsilon(\sum_g a_g g) = \sum_g a_g$. Use this augmentation for the right A-action on k. Then

$$k \otimes_A M \cong M_G := M/\langle gm - m : g \in G,\ m \in M\rangle_k.$$

Indeed the balancing relation identifies $1 \otimes gm$ with $1 \otimes m$, and the inverse sends $1 \otimes m$ to the class of m. The fixed subspace M^G consists of those m satisfying $gm = m$ for every g. For $M = k[S] \otimes_k V$ with V a trivial G-module, the fixed subspace consists of V-valued vectors constant on each orbit. The quotient map to $k[S/G] \otimes_k V$ sums coordinates over each orbit. On a constant vector v on an orbit O this quotient takes value $|O|v$.

Proposition 7.1 (Averaging)**.** *If $|G|$ is invertible in k, the composite $M^G \hookrightarrow M \twoheadrightarrow M_G$ is a natural isomorphism. Its inverse sends $[m]$ to em, where $e = |G|^{-1} \sum_{g \in G} g$. For every M, $\mathrm{Tor}_q^A(k, M) = 0$ for $q > 0$.*

Proof. Both $ge = e$ and $eg = e$ hold by reindexing the sum. Also $e^2 = e$ and e acts as the identity on fixed vectors. Hence $m \mapsto em$ factors over M_G and takes values in M^G. In M_G, each $[gm]$ is $[m]$, so $[em] = [m]$. This proves the isomorphism. The right ideal eA is a summand of the free right module A, via left multiplication by e. The map $k \to eA$ sending 1 to e is an isomorphism of right modules since $ea = \epsilon(a)e$ and $\epsilon(e) = 1$. The right module k is therefore projective. □

7.2 The metric on shared parameters

Over $k = \mathbb{R}$, let V be a finite-dimensional Euclidean space and $P = V^S$ with the product inner product. The permutation representation acts by isometries. Put $U = V^{S/G}$. The tying map $i : U \to P$ sends θ to the vector with $i(\theta)_s = \theta_{[s]}$, where $[s]$ denotes the orbit of s. Its image is P^G. For a differentiable loss $L : P \to \mathbb{R}$, write g_s for the s-component of $\nabla L(i\theta)$. The tied derivative is the pullback $i^* DL(i\theta)$. On $v \in U$ it is

$$D(L \circ i)(\theta)[v] = \sum_{O \in S/G} \left\langle \sum_{s \in O} g_s, v_O \right\rangle.$$

With the ordinary product metric on U, the O-component of the gradient is $\sum_{s \in O} g_s$. The metric induced by i has inner product

$$\langle u, v\rangle_i = \sum_O |O| \langle u_O, v_O\rangle.$$

In the induced metric, the component for O becomes the average of the g_s over O. Let R_G be the Reynolds operator $m \mapsto em$ on P. Then

$$i\nabla_i (L \circ i)(\theta) = R_G \nabla L(i\theta).$$

Indeed for $s \in O$ every index $t \in O$ occurs as $g^{-1}s$ for exactly $|G_s|$ group elements, so R_G averages uniformly over O. This is also the orthogonal projection onto P^G: the operator is idempotent, and reindexing by g^{-1} gives self-adjointness.

For example, let C_2 swap a, b and fix c, and take $V = \mathbb{R}$. The map i sends (θ, η) to (θ, θ, η). For $y = (1, 3, 2)$ and $L(p) = \frac{1}{2} \sum_{s \in S} (p_s - y_s)^2$, the product-metric gradient in U is $(2\theta - 4, \eta - 2)$, while the induced metric gives $(\theta - 2, \eta - 2)$. At the origin the first shared coordinate receives two contributions to the derivative, and the induced metric divides that sum by the orbit size. This difference matters when a single learning rate is used across orbits of unequal sizes.

When the ambient loss is invariant, its ambient gradient at a tied point is itself fixed. Differentiating $L \circ g = L$

at $p \in P^G$ gives $DL(p)[gv] = DL(p)[v]$. Since g acts orthogonally, uniqueness of the Riesz representative gives $g\nabla L(p) = \nabla L(p)$. Consequently the induced gradient agrees with the ambient gradient after inclusion by i, and the product-metric gradient still includes the orbit-size factors. This identifies the metric implicit in restricting Euclidean gradient flow to the tied subspace.

Tying also has a direct architectural consequence. Suppose a parameterized family $F : P \times X \to Y$ satisfies diagonal equivariance, $F(gp, gx) = gF(p, x)$, for given G-actions on X, Y. Each tied parameter $i\theta$ gives an equivariant model, since $g(i\theta) = i\theta$ and thus $F(i\theta, gx) = gF(i\theta, x)$. The hypothesis is a property of the untied family, checkable at its generating operations.

7.3 The two-sided bar and stabilizer homology

The balancing construction is the degree-zero part of a derived tensor product. Kříž and May develop derived tensor products of differential graded modules by cell approximation in Part III, §4, starting on p. 53 [1]. For augmented algebras the two-sided tensor bar, reviewed in Part IV, §6, p. 71, supplies a concrete model. We use its simplicial formula for the associative algebra $A = k[G]$, with homological degrees increasing with bar length. Their operadic comparison in Part II, §4, p. 35 uses a monad bar $B(F, C, A)$ with operadic monad C and a C-algebra A. The bar here uses tensor powers of the group algebra and its ordinary right and left modules.

Explicitly, use $B_q(k, A, M) = k \otimes_k A^{\otimes q} \otimes_k M$ with differential given by the alternating sum of faces. The first face applies the augmentation to the first algebra factor, interior faces multiply two adjacent factors, and the last face acts on M. Degeneracies insert the unit. Replacing M by A gives a free right A-module in every degree. The augmented underlying complex contracts by appending $1 \in A$ as an extra last factor, with sign $(-1)^{q+1}$ in degree q. Indeed the last face of this extra degeneracy is the identity and every earlier face commutes with it in the adjacent degree. Thus $B_\bullet(k, A, A) \to k$ is a free resolution, and tensoring it over A with M gives

$$H_q(B_\bullet(k, A, M)) \cong \mathrm{Tor}_q^A(k, M).$$

The degree-one differential sends a generator $[g] \otimes m$ to $m - gm$, giving the ordinary quotient in degree zero. One can also use normalized chains, since the degenerate subcomplex of a simplicial k-module is contractible.

For a permutation module $M = k[S]$, this complex is the chain complex of the nerve of the action groupoid $G \ltimes S$. For the given face convention, the generator $(g_1, \dots, g_q, s)$ corresponds to the string with terminal object s and successive arrow labels $g_1^{-1}, \dots, g_q^{-1}$, beginning at $g_1 \cdots g_q s$. Deleting the first object drops g_1. Deleting the last replaces s by $g_q s$. Composing adjacent arrows takes g_i, g_{i+1} to $g_i g_{i+1}$ in the bar coordinates. These are exactly the three types of faces above. The groupoid describes how the identification is made, retaining the automorphisms of a representative of each orbit.

Proposition 7.2 (Orbit decomposition)**.** *For a finite G-set S, taking a representative s in each orbit gives*

$$\mathrm{Tor}_q^{k[G]}(k, k[S]) \cong \bigoplus_{[s] \in S/G} \mathrm{Tor}_q^{k[G_s]}(k, k).$$

Proof. The map $g \otimes 1 \mapsto gs$ identifies the induced module $k[G] \otimes_{k[G_s]} k$ with the span of the orbit of s. The translates of s are indexed by left cosets gG_s. If $Q_\bullet \to k$ is a projective right $k[G]$-module resolution, it remains a projective resolution over $k[G_s]$ because $k[G]$ is free on the right over $k[G_s]$. For each orbit, tensor associativity identifies

$$Q_\bullet \otimes_{k[G]} (k[G] \otimes_{k[G_s]} k) \qquad \cong Q_\bullet \otimes_{k[G_s]} k.$$

Taking homology and the sum over orbits proves the formula. □

Let $\mathrm{char}(k) = p$ and $G = C_p = \langle g \rangle$. Writing $u = g - 1$, one has $A = k[u]/(u^p)$, and the augmentation sends u to zero. A free resolution of k alternates multiplication by u in odd degrees and by u^{p-1} in positive even degrees. The kernels of these two multiplications are (u^{p-1}) and (u), respectively, so the images are exactly the next kernels. Their composites vanish since $u^p = 0$. Tensoring with the trivial module k makes both differentials zero, whence $\mathrm{Tor}_q^A(k, k) \cong k$ for every $q \geq 0$.

For the preceding action of C_2 on $\{a, b, c\}$ over $\mathbb{F}_2$, the swapped orbit is free and the fixed slot has stabilizer C_2. Proposition 7.2 gives two degree-zero classes and one class in each positive degree. Directly, on the fixed-slot summand the augmented group algebra has square-zero augmentation ideal (u). In the normalized bar the endpoint actions and every internal multiplication vanish, leaving one copy of k in each bar degree. Thus ordinary orbit pooling accounts for the same number of coordinate classes over both fields, and the chain-valued modular quotient carries additional information about stabilizers. Over $\mathbb{R}$, Proposition 7.1 reduces the derived quotient to ordinary balancing.

For an explicit invariant feature in this modular setting, take the dimensions h_j of the first two homology groups of the quotient complex. They depend only on the isomorphism class of the finite G-set. A single free

C_2-orbit and a single fixed point both have $h_0 = 1$ over $\mathbb{F}_2$. They have $h_1 = 0$ and $h_1 = 1$, respectively, so a real linear readout of (h_0, h_1) can distinguish these two inputs after the homology computation. This example supplies an exact representation statistic for finite symmetry data. Its value as a feature in any larger learning problem depends on whether the target depends on the stabilizers in the input.

8 Compositional AI and related work

8.1 Wiring and differentiable semantics

An operad specifies which interfaces can be substituted into one another. An algebra specifies what a substitution does to data. Vagner, Spivak, and Lerman give this separation a concrete form for open dynamical systems: their wiring diagrams compose syntactically, and an algebra interprets the diagrams by wiring together systems of differential equations [5, Sec. 3 and Prop. 4.5]. In neural programs, the analogous syntax can record port types, parameter occurrences, and which subprograms consume each intermediate value. When a wire feeds multiple consumers, the reverse pass sums all the cotangents returned along that wire. Consequently, substitutions can be compiled while keeping the equality of that local sum with the pullback along the forward wiring map.

Gavranović et al. treat constraints and implementations of deep learning architectures by monads and endofunctor algebras in a 2-category of parametric maps [4]. This accounts for recurrent architectures and gives an algebraic language for equivariance. The chain-complex setting used here specifies multilinear operations and modules over a fixed algebra, so explicit passage to an algebra and its module data is needed when applying it to a smooth parameterized program. For example, a differentiated relation between two substitutions compares their effects on inputs and parameter directions. Such a relation is additional information about which perturbations preserve the algebra.

Fong, Spivak, and Tuyéras give a symmetric monoidal functor from parametrized differentiable functions to learners under an invertibility hypothesis for the scalar error derivative [2, Thm. III.2]. Cruttwell et al. encode reverse derivatives as lenses and thereby treat models, losses, and optimizers as composable data [3, Secs. 2–3]. Spivak gives operadic dynamics of smooth adaptive arrangements, with gradient descent as one of the semantics [8]. Those formulations describe learning dynamics. Here the cotangent pullback is the dual of the linearized substitution. Turning the resulting covector into a parameter update also requires a choice of metric or optimizer.

8.2 A module of probability perturbations

For nonempty finite sets $X_1, \dots, X_n, Y$, a stochastic operation K assigns a probability measure $K(- \mid \boldsymbol{x})$ on Y to each input tuple. To substitute a stochastic operation H_i into slot i, first sample their results independently conditional on their respective input blocks $\boldsymbol{u}_i$. The substituted kernel is

$$(K(H_1, \dots, H_n))(y \mid \boldsymbol{u}) = \sum_{\boldsymbol{x}} K(y \mid \boldsymbol{x}) \prod_i H_i(x_i \mid \boldsymbol{u}_i).$$

Rearranging the finite sums verifies associativity. Permuting inputs reindexes the sums, and identity kernels are units. Taking these kernels as operations therefore defines a colored operad in sets.

Let $A_X = \mathbb{R}^X$ be the vector space of signed measures, $\epsilon_X(a) = \sum_{x \in X} a(x)$, and $M_X = \ker \epsilon_X$. The linearization of the kernel operad acts on A by

$$K_A(a_1, \dots, a_n)(y) = \sum_{\boldsymbol{x}} K(y \mid \boldsymbol{x}) \prod_i a_i(x_i).$$

Since $\sum_y K(y \mid \boldsymbol{x}) = 1$, its output mass is $\prod_i \epsilon_{X_i}(a_i)$. Hence the operation restricts to M_Y whenever a chosen input lies in M_{X_i}. These one-marked-input restrictions satisfy the module axioms by restricting the algebra substitution identities. In particular the first-order change in the result under varying one probability law is the marked-input map on its signed zero-mass perturbation. Baseline laws are nonnegative and normalized. At a boundary point of the probability simplex, a zero-mass direction can be feasible in only one sign.

If M_X is equipped with $\|m\|_{\mathrm{TV}} = \frac{1}{2} \sum_x |m(x)|$, the marked-input map at probability laws in the remaining slots has operator norm at most the diameter $\sup_{\boldsymbol{x}, \boldsymbol{x}'} \|K(- \mid \boldsymbol{x}) - K(- \mid \boldsymbol{x}')\|_{\mathrm{TV}}$. Indeed a nonzero m is $\|m\|_{\mathrm{TV}}$ times the difference of two probability laws obtained by normalizing its positive and negative parts. The two outputs after replacing the marked input are convex mixtures of rows of K. Their distance is bounded by that diameter, by comparing all pairs of rows with the product mixture weights. This is the contraction factor used in the following consistency test.

8.3 Stochastic substitution and answer consistency

Bottman and Richardson model question decomposition by a question operad. Their operations are question tem-

plates, and composition plugs the output of one question into a blank of another. They propose comparing answers across partial evaluations of a decomposition tree [9]. Bottman, Liu, and Richardson report a companion empirical study of this consistency signal on multi-hop question-answering datasets [10]. One mathematical use of an operadic specification is to say which evaluation orders should agree under a particular answer semantics. The following bound quantifies the propagation of local discrepancies for finite stochastic evaluations.

Fix a rooted tree with a nonempty finite answer set X_v at each vertex. For an internal vertex v, let I_v be the product of the answer sets of its children. Two evaluators have local Markov kernels $K_v, \widetilde{K}_v$ from I_v to distributions on X_v. At a leaf I_v is a singleton and the kernels specify its two answer laws. Assume the child evaluations are sampled independently for each evaluator, with original query and external random context fixed. At each vertex the evaluator applies its kernel to the product of those laws. Write $\mu_v, \widetilde{\mu}_v$ for the resulting output laws. For probability laws on a finite set let $\|\cdot\|_{\mathrm{TV}}$ denote total variation distance, defined as the supremum of the absolute difference in probability over all events. Put

$$\delta_v = \sup_{i \in I_v} \|K_v(i) - \widetilde{K}_v(i)\|_{\mathrm{TV}},$$
$$\eta_v = \sup_{i,j \in I_v} \|K_v(i) - K_v(j)\|_{\mathrm{TV}}.$$

Thus $0 \le \eta_v \le 1$.

Proposition 8.1 (Propagation of discrepancies)**.** *For the root r, the propagated discrepancy satisfies*

$$\|\mu_r - \widetilde{\mu}_r\|_{\mathrm{TV}} \le \sum_v \delta_v \prod_{w \in \mathrm{Anc}(v)} \eta_w, \tag{6}$$

where $\mathrm{Anc}(v)$ *consists of strict ancestors and an empty product is one.*

Proof. Let $e_v = \|\mu_v - \widetilde{\mu}_v\|_{\mathrm{TV}}$ and let $\alpha_v, \widetilde{\alpha}_v$ be the two laws on I_v. Independent maximal couplings of the child laws give $\|\alpha_v - \widetilde{\alpha}_v\|_{\mathrm{TV}} \le \sum_u e_u$, summed over children u. Apply K_v to a maximal coupling of the two input laws. When the two inputs agree, couple the output samples to be equal. When they differ, the two conditional output laws have a coupling with mismatch probability at most η_v. This bounds the propagated term by $\eta_v \|\alpha_v - \widetilde{\alpha}_v\|_{\mathrm{TV}}$. Replacing K_v by $\widetilde{K}_v$ at fixed input law adds at most δ_v by the triangle inequality under mixing. Hence $e_v \le \delta_v + \eta_v \sum_u e_u$. Iterating this recurrence gives (6). □

For example, if two leaves perturb their distributions by at most 0.05 each and a common parent uses a kernel with $\eta_v = 0.2$, its output moves by at most 0.02. If each row of the parent kernel also changes by at most 0.01, the bound increases to 0.03. The estimate holds for pairwise comparisons after specifying a common evaluation tree and the kernels being compared. In a shared-child computation or a program that carries conversation state between branches, marginal laws can agree even when joint laws differ. For instance, two fair bits can be independent in one evaluator and always equal in the other. Their marginal laws agree, yet exclusive-or distinguishes the evaluators. In that setting a valid comparison has to track the joint law of the consumed inputs.

The bound compares output laws. A ground-truth predicate, a formal checker, or a task-specific notion of utility supplies a separate test of the answer. A one-colour grammar with a constant answer at every operation can be perfectly consistent and wrong on every question with a different correct answer. Thus tree consistency can diagnose certain difficulties of substitution while soundness still depends on the interpretation of the elementary steps and on the external correctness test.

9 Scope and outlook

The application to smooth architecture evaluation concerns the degree-zero part of the monograph's setting. The passage to $\mathbb{R}[P]$ ensures that the module operators are defined on real chain complexes concentrated in degree zero. It gives a model for organizing derivatives and a check on compositional implementations. The original feature operations may be nonlinear. For a piecewise smooth layer the calculation applies on regions of differentiability. A kink, such as that of ReLU at zero, requires an additional derivative convention.

There are several different meanings of tangent structure. In the smooth calculation, a tangent space $T_x X$ consists of directions in a parameter or state space. In the Quillen-theoretic setting, the tangent ∞-category of a presentable ∞-category at an object is the stabilization of the slice. For operadic algebras under stable hypotheses, its comparison to operadic modules is a theorem about that stabilization [6, Corollary 1.0.5]. This supports the role of operadic modules in homotopical linearization. Applying that theorem to learning requires a model of the architecture or data in the specified category, as well as a definition of the perturbations to be measured. The present paper directly establishes the smooth derivative interpretation and computes a separate derived symmetry example.

The clearest uses of the monograph in machine learning are the interpretation of typed architecture substitutions,

the module law for one-input sensitivities, and the derived analysis of symmetry relations when ordinary balancing loses homology. For AI systems whose components return stochastic outputs, agreement between two routes is an empirical question about suitably typed kernels. The consistency calculation in Section 8 supplies a diagnostic with clear sampling assumptions. The bridge in both deterministic and stochastic cases comes from the marked-slot module and the precise rule for composing it.

Reproducibility. The source bundle contains the full LaTeX and a standard-library Python program for the worked numerical derivative and symmetry checks in Section 6. These checks supplement the proofs and are limited to the worked examples.

Authorship. The public byline is Anonymous. A language-model assistant helped draft, typeset, check, and revise the manuscript under the submitter's direction.

References

[1] I. Kříž and J. P. May. *Operads, algebras, modules and motives*. Astérisque 233, Société Mathématique de France, 1995. https://www.numdam.org/item/AST_1995__233__1_0/. Page locators in this paper follow the authors' 109-page version, https://math.uchicago.edu/~may/BOOKS/KM.pdf.

[2] B. Fong, D. I. Spivak, and R. Tuyéras. Backprop as functor: A compositional perspective on supervised learning. In *Proc. 34th Annual ACM/IEEE Symposium on Logic in Computer Science*, 2019. https://arxiv.org/abs/1711.10455.

[3] G. S. H. Cruttwell, B. Gavranović, N. Ghani, P. Wilson, and F. Zanasi. Categorical foundations of gradient-based learning. In *Programming Languages and Systems*, LNCS 13240, pp. 1–28, 2022. https://arxiv.org/abs/2103.01931.

[4] B. Gavranović, P. Lessard, A. Dudzik, T. von Glehn, J. G. M. Araújo, and P. Veličković. Position: Categorical deep learning is an algebraic theory of all architectures. In *Proc. 41st International Conference on Machine Learning*, PMLR 235, pp. 15209–15241, 2024. https://proceedings.mlr.press/v235/gavranovic24a.html.

[5] D. Vagner, D. I. Spivak, and E. Lerman. Algebras of open dynamical systems on the operad of wiring diagrams. *Theory and Applications of Categories*, 30(51):1793–1822, 2015. https://arxiv.org/abs/1408.1598.

[6] Y. Harpaz, J. Nuiten, and M. Prasma. Tangent categories of algebras over operads. *Israel Journal of Mathematics*, 234:691–742, 2019. https://arxiv.org/abs/1612.02607.

[7] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola. Deep Sets. *Advances in Neural Information Processing Systems* 30, 2017. https://arxiv.org/abs/1703.06114.

[8] D. I. Spivak. Compositional dynamics in learning and mechanics. arXiv preprint, 2026. https://arxiv.org/abs/2606.28984.

[9] N. Bottman and K. Richardson. Operads for compositional reasoning in LLMs. arXiv preprint, 2026. https://arxiv.org/abs/2606.13634.

[10] N. Bottman, Y. Liu, and K. Richardson. Operadic consistency: A label-free signal for compositional reasoning failures in LLMs. arXiv preprint, 2026. https://arxiv.org/abs/2606.13649.

[11] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018. https://jmlr.org/papers/v18/17-468.html.

[12] J. A. Fike and J. J. Alonso. The development of hyper-dual numbers for exact second-derivative calculations. *49th AIAA Aerospace Sciences Meeting*, paper AIAA 2011-886, 2011. https://doi.org/10.2514/6.2011-886.