RESEARCH MANUSCRIPT NOVEMBER 2026

Higher-Categorical Semantics of Machine Learning and Artificial Intelligence

Parametric composition, descent of trained models, and stochastic agents

Anonymous manuscript

Prepared for *Higher Categorical Semantics*

Abstract

We give a categorical semantics for trainable programs that separates parametric composition, reverse differentiation, gauge equivalence, and stochastic execution. In a smooth ∞-topos, an architecture with parameter symmetries defines a quotient stack of predictors. A symmetry-invariant risk defines a relative stationary stack by a homotopy pullback. We identify its families and stabilizers, prove descent, and compute a family of globally defined optimal predictors that admits local weights but no global choice of weights, and a separate gerbe obstruction for an ineffective symmetry. We compare this smooth stack with a derived critical locus, computing the Koszul homology and tangent obstruction of a degenerate loss. We also give precise comparison results for coordinate changes, permutation-equivariant attention, compositional stochastic agents, and federated updates. Six Rust scenes render the loss, decision, covering, and optimization geometries used in the proofs. Notes relate the construction to prior work on learners, reverse differential categories, Markov categories, higher toposes, and derived geometry.

Keywords. Machine learning, reverse differentiation, smooth stacks, derived critical loci, Markov categories, agent semantics.

Preparation. AI assistance was used in research synthesis, mathematical exposition, and the companion software. Anonymous review copy.

Contents

1 What a higher-categorical semantics of learning should preserve 4
1.1 From a program to a family of trained models . . . 4
1.2 Prior work and the comparative claim . . . 4
1.3 The five equivalences . . . 5
1.4 Conventions and structure . . . 6

2 Parametric programs and compositional architecture 7
2.1 Parameterizing a Cartesian category . . . 7
2.2 The semantics of a wiring diagram . . . 7
2.3 A compositional representation of symmetries . . . 8
2.4 Interpreting changes of architecture . . . 9

3 Reverse differentiation and the semantics of learners 10
3.1 The reverse derivative with its types . . . 10
3.2 Reverse composition of parameterized maps . . . 10
3.3 Losses, update maps, and requests . . . 11
3.4 Differentiation of an objective assembled from samples . . . 11

4 The geometry required by an optimizer 13
4.1 A covector is intrinsic . . . 13
4.2 A change of coordinates that changes Euclidean descent . . . 13
4.3 Fisher information and singular parametrizations . . . 14
4.4 When updates descend . . . 14
4.5 Hessian comparisons at a stationary point . . . 15

5 Smooth higher toposes and the moduli of predictors 16
5.1 The site of tests . . . 16
5.2 A symmetry quotient and its points . . . 16
5.3 Evaluation through the quotient . . . 17
5.4 Coherent comparison across architectures . . . 17
5.5 Paths of parameters and isomorphisms of presentations . . . 18

6 The stationary stack as a homotopy pullback 19
6.1 A vertical equation over data . . . 19
6.2 Naturality and its failure for inclusions . . . 20
6.3 Dataset quotients and symmetry . . . 20
6.4 Selecting global minima when the risk permits it . . . 21

7 Descent and an obstruction to global weights 22
7.1 Lifts through a free presentation . . . 22
7.2 A squared one-neuron model . . . 22
7.3 What changes at the fixed point . . . 23
7.4 The overlap signs explicitly . . . 24

8 A degree-two obstruction from ineffective symmetry 25
8.1 From observable families to lifts of their gauge data . . . 25

8.2 The obstruction calculation . . . 25
8.3 A projective family of exact fits . . . 26

9 Derived critical loci and singular losses 27
9.1 Two interpretations of a zero locus . . . 27
9.2 Koszul model and the tangent complex . . . 27
9.3 A quartic loss with nontrivial negative cohomology . . . 28
9.4 Gauge equivalence and derived equations . . . 28

10 Stochastic learning in Markov categories 30
10.1 Kernels and tensor composition . . . 30
10.2 Conditioning and Bayesian inversion . . . 30
10.3 Expectation and differentiation . . . 31
10.4 Stochastic optimizers and latent state . . . 31
10.5 Couplings and shared evaluation . . . 32

11 Attention, sequence models, and causal sampling 33
11.1 Attention as a parameterized layer . . . 33
11.2 Permutation equivariance . . . 33
11.3 The differentiation of an attention kernel . . . 34
11.4 An autoregressive model and a sampling policy . . . 34

12 Open stochastic agents and temporal composition 35
12.1 A finite-step agent interface . . . 35
12.2 Delayed feedback . . . 36
12.3 Simulations preserve finite observations . . . 36
12.4 Path laws and feedback from learning . . . 37

13 Federated learning, descent, and the arithmetic of updates 38
13.1 Data descent and finite sums . . . 38
13.2 One synchronized step . . . 38
13.3 An exact two-step discrepancy . . . 39
13.4 Why averaging and quotienting can disagree . . . 39

14 Observation, expressivity, and composition across levels 40
14.1 The observable image of a model stack . . . 40
14.2 Evaluation on a dataset . . . 40
14.3 Real semantics and implemented finite maps . . . 41
14.4 Approximation and architectural semantics . . . 41
14.5 Limits for coherent training constraints . . . 41

15 A stack of optimizer histories 43
15.1 The graph of a training step . . . 43
15.2 Terminal conditions and preserving presentation data . . . 44
15.3 Histories, limits, and observable transition laws . . . 44
15.4 Transporting optimizer histories . . . 45

16 Six exact geometries rendered in Rust 46

16.1 Two metrics on one loss bowl 46
16.2 A smooth exact XOR surface 46
16.3 The double cover of optimal predictors 48
16.4 Federated quadratic trajectories 48
16.5 The flat bottom of a quartic loss 49
16.6 Antipodal monodromy and the projective plane 50
16.7 Running and reproducing the figures 50

17 Notes on the literature and on the scope of the results **52**
17.1 Composition, learning, and differentiation 52
17.2 Geometry, stacks, and obstruction theory 53
17.3 Probability, sequence modeling, and agency 55
17.4 Distributed training and empirical scope 56
17.5 How to read a claim of equivalence 57

References **58**

1 What a higher-categorical semantics of learning should preserve

1.1 From a program to a family of trained models

A predictor with a specified weight vector is a map from inputs to outputs. A trainable program adds the parameter space, the dependency of the predictor on the parameters, the objective, and the operation that selects or updates parameters. A deployed agent may also carry state, sample actions, and use observations to decide whether to update. There is no single equivalence relation that preserves all of these aspects. Two weight assignments can give the same predictor, have different optimizer state, and interact differently with the same environment after training resumes. A useful semantics makes the chosen level of equivalence explicit.

We start with a smooth parameterized map

$$f : P \times X \longrightarrow Y.$$

Here X and Y are input and output types and P is the parameter type. A task adds observations and a loss. The reverse derivative of the loss with respect to P is a covector. An optimizer requires additional data for turning that covector into a displacement or a probabilistic update. An action of a group G can identify parameter settings that express the same predictor. If one studies a family of such models over a test manifold, the compatible changes of local weights can carry topology. The higher topos is the category in which these families and their gluing isomorphisms are objects, and the stationary equation can be imposed by a limit.

This approach addresses two uses of higher structure. The ordinary smooth quotient stack remembers isomorphisms, stabilizers, and descent in smooth families. Its values are groupoids for the ordinary Lie-group actions used here. Separately, the derived critical locus retains relations in the equation and an obstruction space for infinitesimal deformations. It is defined in a derived category of test objects. The two refinements can be combined with hypotheses on the action and the derived site. Neither refinement is obtained simply by replacing a real parameter vector by an abstract space of models.

1.2 Prior work and the comparative claim

The parameterized map and its compositional treatment have direct predecessors. Fong, Spivak, and Tuyéras show how backpropagation and supervised learning can be organized as a functor of parameterized maps [1]. Fong and Johnson compare learners to lenses [2]. Cruttwell, Gavranović, Ghani, Wilson, and Zanasi organize gradient-based learning through reverse differential structure, parameterization, and

lenses [3]. Reverse differential categories axiomatize the reverse chain rule [4], and Gavranović's thesis gives a broader categorical account of parameterization and deep learning [5]. These are the principal antecedents for Sections 2–4. The surveys by Shiebler, Gavranović, and Wilson and by Jia and coauthors place this literature beside categorical probability, invariance, and topos-based approaches [9, 10].

The geometric part combines this input with standard construction principles from higher topos theory, stack descent, and derived geometry [33, 34, 36, 41, 42]. The contribution is a precise comparison in one learning model. The same invariant objective yields a stationary quotient stack, a derived critical refinement on polynomial charts, and a coordinate-independent vector field after a metric is chosen. These structures have different tests for equality and descent. We prove these comparisons and calculate examples that separate them. In particular, an entire family of globally optimal predictors can be well defined even though no continuous global choice of the given weights realizes it. The proof uses a concrete square-cover obstruction and a loss whose global minimizers can be identified exactly.

The stochastic and operational parts use Markov categories, probability kernels, and coalgebraic systems [47, 46, 61]. They distinguish state transitions, externally observable histories, and coherence of architectural comparison. Attention, reinforcement learning, and federated optimization enter as typed constructions and examples. Claims about a category of agents depend on the interface, time convention, and chosen observation criterion. Claims about performance or generalization require statistical assumptions in addition to a compositional semantics.

1.3 The five equivalences

The manuscript uses several comparison relations. They are summarized here to specify what a later theorem actually identifies.

Level	Data preserved by the comparison
Parametric program	The forward map after an explicit parameter change.
Differentiated program	The forward map and transport of input and parameter covectors.
Smooth model stack	Families, their symmetry arrows, stabilizers, and effective descent.
Derived stationary locus	Infinitesimal solutions and relations among equation residuals on derived tests.
Stochastic agent behavior	Output laws at the selected interfaces under the stated input or environment protocol.

An optimizer comparison additionally requires compatibility of metric, state, and discretization. Invariance of the objective under a reparametrization determines the cotangent transformation rule. It does not determine a coordinate update. Conversely a change in the selected observations can identify agents that differ internally even when no equivalence of their parameter types exists.

1.4 Conventions and structure

All main differentiation results use finite-dimensional smooth Euclidean spaces or ordinary smooth manifolds. Smooth test spaces and measured state spaces are introduced separately. ReLU, floating point, and discrete code appear only with their own semantics. A polynomial example is used when an exact derived-algebraic computation is needed. In the stochastic sections, kernels on standard Borel spaces are the concrete models. Finite horizons are used for the main operational composition claims.

Sections 2–4 establish parameterized composition, reverse differentiation, and optimizer geometry. Sections 5–9 give the higher-topos and derived constructions, including the global obstruction. Sections 10–14 extend the account to stochastic models, attention, agents, and distributed training. Section 16 gives exact geometric examples and their Rust viewer. The final notes relate each construction to its closest published precedents and state its hypotheses.

2 Parametric programs and compositional architecture

2.1 Parameterizing a Cartesian category

Let $\mathcal{C}$ be a category with finite products. For a first model take finite-dimensional real vector spaces as objects and smooth maps as arrows. A parameterized arrow $X \rightsquigarrow Y$ is a pair (P, f) with $f : P \times X \to Y$. A reparametrization $(P, f) \Rightarrow (Q, g)$ is a map $h : P \to Q$ satisfying

$$f = g \circ (h \times \mathrm{id}_X). \tag{2.1}$$

The direction of the arrow records how a parameter for the source program is interpreted as a parameter for the target program. An inclusion of a constrained parameter subspace is a legitimate arrow even if it has no inverse. For a symmetry of a presentation one instead uses an invertible reparametrization. This distinction affects the groupoid of model families.

For $(P, f) : X \rightsquigarrow Y$ and $(Q, g) : Y \rightsquigarrow Z$, take parameter $P \times Q$ and forward map

$$((p, q), x) \longmapsto g(q, f(p, x)). \tag{2.2}$$

The identity uses the terminal parameter and the identity of the interface. Horizontal composition of reparametrizations is their product. Equation (2.1) for the composite follows by applying the two reparametrization equations in turn. The product associator and unitors furnish the coherent comparisons for repeated composition. This gives the usual bicategorical Para construction in this Cartesian case [3, 5]. One can strictify the bookkeeping by using lists of parameter blocks.

Parallel composition uses a product of interfaces and a product of parameter spaces. Symmetries of those products exchange independent components. When a component is reused, care is needed about the meaning of the product. Using the same parameter twice means precomposing with the diagonal on the parameter space. It is a change of parameterization. Using two independent copies of a parameterized arrow allows two different parameters. Their joint space agrees with the first model only after restriction to the diagonal.

2.2 The semantics of a wiring diagram

An acyclic circuit of composable smooth boxes can be interpreted by products, projections, and composition. Copying an ordinary value is the diagonal $X \to X \times X$ and discarding it is the unique map to a terminal object. An input used twice is copied explicitly at the level of the diagram. The free Cartesian category on a signature of typed boxes presents this wiring. Specifying an interpretation for each generator determines the interpretation of each circuit modulo the Cartesian equations [12].

A layer may be linear, smooth nonlinear, or itself parameterized. Consider $f((W, b), x) = Wx + b$. A smooth activation σ applied coordinatewise is a parameter-free arrow. A depth-two circuit then has parameter space (W_1, b_1, W_2, b_2) with the expected composite forward map. A change of width is a change of the type of the intermediate interface. The external predictor may remain the same under an explicit embedding of the parameters, as when an added hidden unit has zero outgoing weight. Such a comparison need not be invertible and does not give a symmetry of a fixed architecture.

A residual block $x \mapsto x + f(p, x)$ can be built from copying, the block f, and addition. A shared-weight recurrent program on a fixed time horizon is built by composing step maps and then using a diagonal that identifies the weight blocks. This distinguishes the finite unrolling of a recurrent system from a feedforward program with unrelated weights at each step. Parameter tying will also determine where the reverse pass adds cotangents.

2.3 A compositional representation of symmetries

Take a hidden layer

$$x \longmapsto W_2\, \sigma(W_1 x + b_1) + b_2.$$

For any permutation matrix S of the hidden width, coordinatewise activation satisfies $\sigma(Sz) = S\sigma(z)$. Thus the parameter transformation

$$(W_1, b_1, W_2, b_2) \longmapsto (SW_1, Sb_1, W_2 S^{-1}, b_2) \tag{2.3}$$

leaves the forward map invariant. Such matrices form a finite group acting on the parameters. In more general architectures an activation may commute with a subgroup of scalings or linear coordinate changes, but this must be checked at the level of its forward operation. Parameter symmetries arise from an equation in the program, not merely from an informal similarity between the weights. Work on neural network symmetries and weight alignment studies their operational consequences [27, 31, 28].

The reverse implication from functional equality to membership in a particular orbit generally fails. For example in a network with zero output weights, many changes of hidden weights have no effect on the output. The subgroup generated by hidden permutations does not relate all such parameter settings. We will therefore keep a map from the symmetry quotient to the predictor sheaf, with no general identification of the two. The chosen symmetry encodes intended changes of presentation. The predictor semantics encodes all functional equalities under the given observation of inputs and outputs.

2.4 Interpreting changes of architecture

Suppose $i : P \to Q$ embeds a narrow model in a wider model without changing its predictor. Equation (2.1) expresses preservation of evaluation. It does not imply that optimization in Q preserves $i(P)$. The new parameter directions can have nonzero gradients, or optimizer noise can move out of the embedded subspace. For transfer of a training result one can compare the restriction of risk and its derivative along i. For transfer of training dynamics one needs tangency and a compatible update operation. This is an instance of a general rule used below: every layer of the semantics needs its own naturality condition.

A family of architectures can be indexed by a small category or higher category K. A morphism can be a permitted inclusion, an interface translation, or a documented equivalence of presentations. We use only coherent comparisons that are supplied as data. For example, preserving the evaluation arrow on a chain of architecture changes supplies commuting triangles. If different chains represent equivalent comparisons, their chosen coherences determine how a diagram of families acts. The higher diagram environment in Section 5 describes which gluing statement follows from this input.

3 Reverse differentiation and the semantics of learners

3.1 The reverse derivative with its types

For a smooth map $f : X \to Y$ between Euclidean spaces, write

$$R[f](x, \eta) = Df_x^* \eta, \qquad \eta \in T^*_{f(x)} Y.$$

The star is the linear dual, with the Euclidean transpose as its matrix in standard coordinates. The chain rule gives

$$R[g \circ f](x, \zeta) = R[f](x, R[g](f(x), \zeta)). \tag{3.1}$$

On manifolds the base point in each cotangent fiber is part of the type. A pullback of a cotangent vector is intrinsic even though an expression for its components uses a local chart. On Euclidean spaces one can suppress the cotangent bundle notation because the bundles trivialize.

A lens from (X, X') to (Y, Y') is a forward map $X \to Y$ and a backward map $X \times Y' \to X'$. In this concrete setting a reverse differentiable map gives the lens $(f, R[f])$ with the primed spaces dual to the corresponding tangent types. Equation (3.1) is exactly the composition rule for these lenses. In a manifold setting a dependent version is needed, since the backward input belongs to a fiber over the forward output. Reverse differential categories give axioms and a generalized formulation [4, 3]. Functional presentations of automatic differentiation give related ways to derive the reverse rule from compositionality [16, 17, 18].

3.2 Reverse composition of parameterized maps

For the composite (2.2), let ζ be an output covector. Reverse through $g(q, y)$ first with $y = f(p, x)$:

$$(\beta_q, \eta_y) = Dg^*_{(q,y)} \zeta.$$

Then reverse through f:

$$(\alpha_p, \xi_x) = Df^*_{(p,x)} \eta_y.$$

The resulting backward map on the whole circuit returns $(\alpha_p, \beta_q, \xi_x)$ under the canonical cotangent splitting for the product. This is exactly the dual of the derivative of the composite. The parameter cotangent for each block is obtained without requiring its full Jacobian as an explicitly stored matrix.

If a parameter p is used in two blocks, the forward parameterization includes a diagonal $\Delta : P \to P \times P$. Its differential is $v \mapsto (v, v)$, and its dual is $(\alpha, \beta) \mapsto \alpha + \beta$. Thus the two parameter cotangents add. The same argument gives addition at a branch of

the forward data-flow graph when both outputs contribute to the loss. Fan-out in the forward direction becomes summation in the backward direction. Discarding a forward output gives the zero covector in the backward direction. These identities explain the familiar accounting rules for gradients at reused variables.

Proposition 3.1 (Compatibility with reparametrization)**.** *Let $f(p,x) = g(h(p),x)$ as in (2.1). For an output covector η, the parameter and input cotangents satisfy*

$$R_P[f](p,x,\eta) = Dh_p^* R_Q[g](h(p),x,\eta), \qquad R_X[f](p,x,\eta) = R_X[g](h(p),x,\eta).$$

Proof. Differentiate $(p,x) \mapsto (h(p),x)$. The tangent map is the block sum of Dh_p and the identity on T_xX. Apply the dual chain rule and project to the two summands. □

The proposition does not require h to be invertible. If h is an immersion, it says the parameter cotangent on the smaller model is the restriction of the larger one along the tangent image. If Dh_p fails to be injective, components in its kernel are invisible to the forward differential. Neither case specifies a preferred Euclidean gradient vector on the smaller model.

3.3 Losses, update maps, and requests

For a target y and smooth loss $\ell : Y \times Y \to \mathbb{R}$, the cotangent supplied at prediction $\widehat{y} = f(p,x)$ is $\partial_1 \ell(\widehat{y}, y)$. Applying the reverse derivative gives $d_p(\ell \circ f)$ and $d_x(\ell \circ f)$, where the target is held fixed. Gradient descent uses the first after choosing a metric and a step rule. The second can be used to compose a learning interface with an upstream block. The distinction between a propagated covector and a requested replacement value matters: converting a covector to a request in the original type requires structure supplied by the choice of loss or update convention.

Fong, Spivak, and Tuyéras construct a functor to a category of learners with implementation, update, and request components, under their hypotheses on error functions and gradient descent [1]. A learner of that kind contains more operations than the reverse lens alone. Cruttwell and coauthors make the separate loss, optimizer, and parameter structure explicit [3]. Here the reverse map is a compositional calculus for covectors. The optimizer is a different arrow, possibly stateful or stochastic. To claim it descends under symmetries one checks its own equivariance.

3.4 Differentiation of an objective assembled from samples

For a finite ordered dataset $D = ((x_i, y_i))_{i=1}^n$ and fixed coefficients a_i, set

$$L(p, D) = \sum_{i=1}^{n} a_i \, \ell(f(p, x_i), y_i).$$

By linearity of the differential,

$$d_p L = \sum_i a_i \, R_P[f](p, x_i, \partial_1 \ell(f(p, x_i), y_i)).$$

This is a finite algebraic equality. It applies before taking expectations or limits. If all weights are equal, permuting the records and preserving multiplicity leaves the risk invariant. If records are weighted, symmetry acts on data and weights together. Repeated records contribute with their multiplicity, which is why passing to a set of examples can change the objective.

A penalty $R(p)$ adds its cotangent to $d_p L$. It can break parameter symmetry even when the sample risk is invariant. To define a stationary stack for a regularized learner, one uses the full invariant risk or reduces the group to the symmetries of that risk. This condition is independent of any optimizer state.

4 The geometry required by an optimizer

4.1 A covector is intrinsic

Let $L : P \to \mathbb{R}$ be a smooth risk and $h : P \to Q$ a diffeomorphism with $L = \widetilde{L} \circ h$. Then $dL_p = Dh_p^*(d\widetilde{L}_{h(p)})$. A metric g on P identifies cotangents with tangents by the musical isomorphism $g_p^\sharp : T_p^*P \to T_pP$. This gives $\nabla_g L = g^\sharp dL$. The construction depends on a metric, but it is natural when the metric is transported along the coordinate change.

Theorem 4.1 (Metric equivariance)**.** *Suppose $h : (P, g) \to (Q, \widetilde{g})$ is an isometry and $L = \widetilde{L} \circ h$. Then*

$$Dh_p \nabla_g L(p) = \nabla_{\widetilde{g}} \widetilde{L}(h(p)).$$

Consequently gradient flows are intertwined on their common interval of existence. Discrete updates are intertwined whenever their retractions or other displacement rules are intertwined by h as well.

Proof. For $v \in T_{h(p)}Q$ the isometry identity gives

$$\widetilde{g}(Dh \nabla_g L, v) = g(\nabla_g L, Dh^{-1}v) = dL(Dh^{-1}v) = d\widetilde{L}(v).$$

Nondegeneracy proves the formula. The flow comparison follows by uniqueness for the corresponding vector field, and the discrete comparison follows by applying the displacement rule to the intertwined tangent vectors. □

A Riemannian exponential is an example of an isometry-compatible displacement rule on its domain of definition. Coordinate addition $z \mapsto z - \eta v$ in two charts need not be intertwined by a nonlinear coordinate change. Indeed Taylor's theorem adds a term of order η^2 when the chart has nonzero second derivative. Natural-gradient vector fields can therefore be invariant under a larger class of reparametrizations than a particular finite step using coordinate addition [22, 23, 24, 25].

4.2 A change of coordinates that changes Euclidean descent

On $P = \mathbb{R}^2$ put $L(x, y) = (x^2 + 4y^2)/2$. Use a second chart $(u, v) = (2x, y/2)$. The same objective expressed in that chart is $\widetilde{L}(u, v) = (u^2/4 + 16v^2)/2$. Euclidean gradient descent in physical coordinates multiplies the two coordinates at each step by $1 - \eta$ and $1 - 4\eta$. Euclidean descent in the second chart, mapped back into physical coordinates, multiplies them by $1 - \eta/4$ and $1 - 16\eta$. For $0 < \eta < 1/8$ both iterations are stable and generally distinct. The derivative comparison remains exactly true. The optimization difference is that the two Euclidean metrics describe different geometries under the identification.

Push the original metric into the second chart. Its matrix is $\mathrm{diag}(1/4, 4)$, so its inverse applied to $(u/4, 16v)$ gives $(u, 4v)$. Since the chart change is linear, coordinate addition now intertwines the discrete updates. The viewer in Section 16 draws the trajectories on the same bowl to make the comparison in a common parameter space.

4.3 Fisher information and singular parametrizations

For a regular statistical family $p \mapsto q_p$, the Fisher form can be written in local coordinates as

$$I_p(v, w) = \mathbb{E}_{q_p}\big[(D_v \log q_p)(D_w \log q_p)\big],$$

when a common dominated representation and the required differentiability hold. This is a covariant bilinear form. If the family is regular and identifiable at the differential level, the form is positive definite and its inverse turns the risk differential into a natural gradient [22, 23]. A redundant parameterization can give null directions. For example, if a model depends only on $s = p_1 + p_2$, the direction $(1, -1)$ is invisible to the Fisher form.

On a chart of constant rank with a smooth quotient $q : P \to Q$ and $I = q^* \widetilde{I}$ for a positive metric on Q, a risk of the form $L = \widetilde{L} \circ q$ determines a natural-gradient vector downstairs. Upstairs the equation $I(v, -) = dL$ determines a class of vectors modulo the kernel. Choosing a horizontal distribution chooses a representative of the class. A matrix pseudoinverse computed using a fixed coordinate Euclidean inner product encodes such an additional choice. Its representative should not be called canonical under arbitrary chart changes. At rank-changing points the existence of a smooth quotient chart is a separate issue.

4.4 When updates descend

Suppose G acts on a parameter manifold and preserves the risk. If the action also preserves g, Theorem 4.1 makes the gradient vector field equivariant. If a discrete optimizer carries auxiliary state S, give S the required G-action and use a map $U : P \times S \to P \times S$. An equivariant U induces a map of quotient stacks by applying U to the equivariant map carried by a principal bundle. This includes algorithms that transform momentum as a tangent or cotangent vector under a symmetry. For a gradient using a product of coordinatewise running moments, one must check the transformation of those moments under the particular group. Permutations act on diagonal statistics by permuting coordinates. General rotations do not preserve the class of diagonal matrices.

Continuous gradient flow, a particular finite-step update, and an update with optimizer memory can thus descend for different groups even with the same risk. At a fixed point

of the group action equivariance imposes additional constraints. For instance under $p \mapsto -p$, any deterministic equivariant update of p alone sends 0 to 0. A randomized update can send 0 to a sign-symmetric law. These are distinct notions of equivariance because their codomains are different.

4.5 Hessian comparisons at a stationary point

At a critical point the Hessian transforms by congruence. If p is critical for L, then

$$\mathrm{Hess}_p L = Dh_p^* \,\mathrm{Hess}_{h(p)} \widetilde{L}\, Dh_p.$$

The usual second derivative of a composite also contains a term with the second derivative of h. That term is multiplied by the first derivative of $\widetilde{L}$ and vanishes at a critical point. Rank and signature of the symmetric bilinear form are therefore invariant under invertible change of coordinates. Its eigenvalues with respect to Euclidean metrics assigned independently in both coordinate systems can differ. In the quadratic bowl above the matrices are $\mathrm{diag}(1, 4)$ and $\mathrm{diag}(1/4, 16)$ in the two charts. Their ratios of eigenvalues are four and sixty-four. With the metric transported, the corresponding metric-raised Hessians are conjugate and have identical spectra.

This comparison explains which geometric statements about the same critical point survive parameter symmetry. An invertible change of coordinates preserves whether the critical point is a nondegenerate minimum, while the rate predicted for a coordinate Euclidean iteration can change. A derived tangent calculation uses the bilinear form as a map to the cotangent space. A convergence-rate calculation usually also selects a metric and step. The distinction between rank and condition number is useful in interpreting geometric analyses of flatness and learning dynamics [27, 23].

5 Smooth higher toposes and the moduli of predictors

5.1 The site of tests

Let Mfd be a small representative in a fixed universe of smooth Hausdorff second-countable manifolds, with the open-cover topology. Set $\mathcal{H} = \mathrm{Sh}_\infty(\mathrm{Mfd})$, the ∞-category of sheaves of spaces. A manifold represents a sheaf, and evaluation at a test T is mapping out of its representable. The subcategory of set-valued sheaves embeds as the 0-truncated objects. Groupoid-valued stacks embed as 1-truncated objects. For these truncated objects ordinary stack descent also gives hyperdescent [33, 34, 35]. The general ∞-topos supports higher limits and genuinely higher-valued objects when those are part of the chosen model.

A parameter family over T is a smoothly varying parameter assignment, possibly with local presentations and transition symmetries. The parameter T is external to the predictor input X. A function in the internal object $\mathcal{F} = Y^X$ has at T the space

$$\mathrm{Map}(T, \mathcal{F}) \simeq \mathrm{Map}(T \times X, Y).$$

When X, Y are ordinary manifolds, this is the set of jointly smooth maps. It need not be represented by a finite-dimensional manifold. The internal exponential gives a model of smooth predictor families without choosing a topology on a vector space of all functions.

5.2 A symmetry quotient and its points

Fix a smooth parameter manifold P, a Lie group G acting smoothly on the left, and an invariant evaluation $f : P \times X \to Y$. Form the quotient stack

$$\mathcal{M} = [P/G] = \mathrm{colim}(P \leftleftarrows G \times P \;\substack{\leftarrow\\\leftarrow\\\leftarrow}\; G^2 \times P \;\cdots) \tag{5.1}$$

in $\mathcal{H}$, where the diagram is the action groupoid's simplicial nerve. The colimit may be constructed by stackifying the action groupoid presheaf. The stage T of the result is the groupoid of smooth right principal G-bundles $E \to T$ with a smooth map $\psi : E \to P$ satisfying $\psi(eg) = g^{-1}\psi(e)$. Arrows are equivariant bundle isomorphisms preserving ψ.

One can derive this description by choosing local sections s_i of a bundle over a cover. They give parameter maps $p_i = \psi(s_i)$. On overlaps $s_j = s_i g_{ij}$, hence $p_j = g_{ij}^{-1} p_i$, and the transition maps satisfy $g_{ij} g_{jk} = g_{ik}$. Conversely these data glue a principal bundle and its equivariant map. Changes of local sections give the expected gauge isomorphisms. This is the same descent mechanism as the classification of ordinary principal bundles, with the additional map to P. General principal ∞-bundles give an appropriate extension of the classifying language [36, 37].

At the test point the bundle is trivial, so the groupoid is the usual action groupoid of points. Its components are orbits, and the automorphism group at p is the stabilizer G_p. For general T a choice of a global weight map $T \to P$ presents the objects whose underlying bundle has been trivialized. Quotient stack points can exist with no such presentation.

5.3 Evaluation through the quotient

By currying, f gives a map $P \to \mathcal{F}$. Invariance makes it a map constant on the arrows of the action groupoid, hence it factors through $\mathcal{M}$. At a stage T the evaluation of (E, ψ) is defined on $T \times X$ by choosing any e over the test point and evaluating $f(\psi(e), x)$. The result is independent of e. Local smooth sections of E show that the descended map is smooth.

If $k : T \to \mathcal{F}$ is a prescribed predictor family, the fiber of $\mathcal{M} \to \mathcal{F}$ over k is the groupoid of realizations of that family in the chosen symmetry presentation. It can be empty. It can have multiple components even when nonempty, since functional equality can identify parameters outside the chosen orbits. It can also have automorphisms because $\mathcal{F}$ is 0-truncated. Consequently evaluation can lose both object and arrow information.

5.4 Coherent comparison across architectures

Let K be a small ∞-category of architecture indices. The category $\mathrm{Fun}(K, \mathcal{H})$ is an ∞-topos. To see this, write $\mathcal{H}$ as a left exact accessible localization of a presheaf category. Applying $\mathrm{Fun}(K, -)$ gives the corresponding objectwise localization of the presheaf category on the product of the presenting category with K^{op}. Limits and equivalences of diagrams are computed objectwise. A coherent diagram of architecture stacks together with evaluations to a compatible diagram of interfaces can therefore be studied internally.

For example, with fixed external interface and evaluation sheaf $\mathcal{F}$, a diagram may take values in the slice over the constant diagram of $\mathcal{F}$. A morphism in K then carries a comparison over the predictor semantics, with any prescribed higher coherences. If one also has a diagram of loss covector objects and natural sections, stationarity can be imposed by an objectwise homotopy limit. The existence of such a natural diagram is a hypothesis. The counterexample for embeddings in Section 6 shows that a forward architectural comparison does not automatically induce a map on stationary loci.

There are two index directions in this account. Passing to an open of T changes the locality of a family. Moving along an arrow of K changes its architectural presentation. Descent in the former does not assert any surjectivity in the latter. The coherent

functor category keeps these directions available without assuming they are the same operation.

5.5 Paths of parameters and isomorphisms of presentations

A smooth path in P is a smooth family over an interval, or over an open interval extended at the endpoints when working on manifolds without boundary. Such a family may connect parameters that implement different predictors. At the test point the arrows in $[P/G](*)$ are the specified gauge isomorphisms. They relate a parameter only to its G-orbit. A path through the parameter manifold need not give such an arrow. Conversely a discrete permutation can give an arrow at the test point without having a continuous path in the symmetry group. A topology or shape functor applied to the smooth stack can record additional paths, but this is an operation on the ambient model, with its own comparison maps.

One can keep the time direction as a smooth test and keep the symmetry direction as arrows. For instance a parameterized curve of predictors has jointly smooth evaluation on the time interval and the input manifold. Descent along an open cover of the time interval matches local curves by symmetry isomorphisms on overlaps. Closing that interval into a circle can introduce a nontrivial principal bundle, even though all local data have smooth representatives. This distinction is the geometric input for the obstructions in Sections 7 and 8.

6 The stationary stack as a homotopy pullback

6.1 A vertical equation over data

Let D be a finite-dimensional smooth dataset manifold, for instance $D_n = (X \times Y)^n$. Let $L : P \times D \to \mathbb{R}$ be a smooth risk invariant under the G-action on P, with G acting trivially on D. Write $B = [P/G] \times D$. The parameter-covector stack over B is

$$V = [(T^*P \times D)/G] \longrightarrow B$$

using the cotangent lift of the action. In particular the base of a covector and the data remain part of its type. There are sections $s, 0$ of $V \to B$, respectively $d_P L$ and the zero covector. Invariance gives the required equivariance:

$$d_P L(gp, d) = \left(D(g^{-1})_{gp}\right)^* d_P L(p, d).$$

The derivative is taken only in P. A moving dataset parameter can therefore be held fixed during the learning equation even though the family varies smoothly in D.

Define the stationary stack as the equalizer of these sections in the slice over B, or equivalently as the homotopy pullback

$$\mathcal{S}_L = B \overset{h}{\underset{V}{\times}} B \tag{6.1}$$

with the two arrows $s, 0$. In the slice the two copies of B are terminal objects over B. Because $V \to B$ has relatively discrete fibers, the witness of equality of covectors over a fixed base object is unique when it exists. Over varying base objects all of their compatible symmetry isomorphisms remain. Taking the pullback in the ambient topos gives the same object over B, with equivalence between the two projected base objects supplied by the pullback comparison.

Theorem 6.1 (Stationary families)**.** *Let Z_L be the sheaf of pairs $(p, d) : T \to P \times D$ whose vertical covector $d_P L$ vanishes. Then $\mathcal{S}_L \simeq [Z_L/G]$. At a test T it classifies principal G-bundles equipped with an equivariant family of stationary parameters and a smooth dataset family on T. It satisfies descent on the test site. Fixing or changing the dataset by pullback commutes with this construction.*

Proof. Apply $\mathrm{Map}(T, -)$ to (6.1). An object of the pullback includes an isomorphism from the value of the derivative section to a zero covector. Transport its second base object along the induced base isomorphism. The derivative covector over the first object must then be zero. Conversely a stationary object together with the identity gives a pullback object. The arrows are exactly the arrows of the quotient that preserve the stationary data. Equivalently, on each local trivialization of the principal bundle

the parameter map lands in the zero-locus sheaf. Such maps glue precisely with the usual bundle transitions. This identifies $\mathcal{S}_L$ with $[Z_L/G]$. The sheaf and stack claims also follow from closure of $\mathcal{H}$ under limits, and the assertion on datasets follows from associativity of pullbacks. □

The sheaf Z_L can have a singular underlying subset. The theorem does not require it to be a smooth submanifold. It is defined by maps into the parameter and dataset manifolds satisfying a smooth equation. At the test point the objects are stationary parameter assignments. Distinguishing local minima, saddles, and maxima requires further information about the risk. A pullback by the zero section imposes stationarity without optimizing globally.

6.2 Naturality and its failure for inclusions

If $h : P \to Q$ is a diffeomorphism and $L_P = L_Q \circ (h \times \mathrm{id}_D)$, then Dh^* is an isomorphism and identifies the respective stationary conditions. With a compatible group action this is an equivalence of the stationary stacks. An inclusion of parameter spaces does not have this property. Consider the risk $L_Q(q_1, q_2) = q_1^2 + q_2$ and $h(p) = (p, 0)$. The restricted risk is stationary at $p = 0$. At $(0, 0)$ the derivative of the larger risk is $dq_2 \neq 0$. An architecture can therefore have a stationary fit when compared with its own allowed variations and fail stationarity in a larger architecture. The example uses an ordinary smooth loss. Adding a constant or using a nonnegative variant, such as $q_1^2 + (q_2 + 1)^2$, gives the same conclusion.

For an embedding, a comparison in the opposite direction can sometimes be stated. If a stationary point of the larger risk lies in the image, its pullback is stationary in the smaller problem by the chain rule. More generally one can form the larger covector restricted to the embedding and then take the fiber of its image under Dh^*. This expresses the smaller stationarity test on the larger cotangent data. It is the correct categorical operation when normal variations remain unconstrained.

6.3 Dataset quotients and symmetry

The ordered dataset manifold has an action by the permutation group of its record slots. A permutation-invariant empirical risk can therefore be defined over the quotient stack of datasets as well. The construction then works over $[D_n/S_n]$ and keeps the automorphisms of a dataset with repeated records. These automorphisms are not model symmetries, but they may appear in the same total family over data. Fixing a dataset as a specific object removes its external variation. In contrast, passing to a space of empirical measures can identify ordered datasets with different multiplicities

or sample sizes after normalization. Whether this is appropriate depends on the risk and the intended statistical interpretation.

6.4 Selecting global minima when the risk permits it

For an ordinary smooth family over a test T, one can define the minimizing subsheaf by requiring each $p(t)$ to minimize $L(-, d(t))$ over the whole parameter manifold. The global comparison is over candidate weights in that fiber of the training problem. It is local in the test variable t: if the property holds for all t on a cover, it holds over the union. Hence this condition defines a subsheaf of the parameter-and-data sheaf and passes to the same symmetry quotient. It can be empty even when the stationary sheaf is nonempty. It can also have no smooth local section near a data value where the preferred minimum jumps.

For a fixed dataset, if the loss has a known lower bound m that is attained, minimizing is equivalent to the single equality $L = m$. For a family with a smooth known optimal-value function one can impose the analogous equality over the dataset base. These are useful special cases because the equality is between morphisms of smooth sheaves. In the squared-neuron examples below the lower bound is zero and every pointwise task attains it. The minimum family is therefore described by vanishing of all residuals. This identification makes it possible to prove the lift obstructions without any assumption about convergence of a training algorithm.

7 Descent and an obstruction to global weights

7.1 Lifts through a free presentation

Suppose a Lie group acts freely and properly on a manifold P, and let $Q = P/G$ with its usual smooth quotient structure. The quotient map is a principal bundle, with a right-action convention chosen as needed. Then the quotient stack $[P/G]$ is represented by Q. In fact a map $q : T \to Q$ determines the pullback of the quotient bundle and its tautological equivariant map to P. Conversely any object of $[P/G](T)$ gives such a map, and its own bundle identifies with that pullback. The identification follows locally by choosing sections and using freeness and transitivity on each orbit.

The representability of the quotient does not imply that every family admits a global map into the atlas P. A global lift $p : T \to P$ is a section of the pullback principal bundle. Local lifts exist by local triviality. If local lifts p_i are chosen, their transitions define a Cech cocycle with values in the sheaf of smooth G-valued functions. The bundle is trivial precisely when this cocycle is a coboundary. Thus a globally defined quotient model can lack global weights in the chosen presentation. This is a failure of global lifting through an atlas, not a failure of descent for the quotient object [34, 36].

7.2 A squared one-neuron model

Take input $x \in \mathbb{R}^2$, weight $w \in \mathbb{R}^2$, and predictor

$$f_w(x) = (w^{\mathsf{T}} x)^2.$$

The sign group $\{\pm 1\}$ acts by $w \mapsto -w$ and preserves evaluation. The corresponding observable is the positive semidefinite matrix $Q = ww^{\mathsf{T}}$. It is determined by its three quadratic evaluations at

$$x_1 = (1, 0), \qquad x_2 = (0, 1), \qquad x_3 = (1, 1)/\sqrt{2}.$$

Indeed the first two give Q_{11} and Q_{22}, and the last gives their half-sum plus Q_{12}. Two nonzero weights give the same matrix exactly when they differ by sign. The zero weight is fixed by the full sign group.

For $\theta \in \mathbb{R}/2\pi\mathbb{Z}$ define

$$Q_\theta = \frac{1}{2} \begin{pmatrix} 1 + \cos\theta & \sin\theta \\ \sin\theta & 1 - \cos\theta \end{pmatrix}. \tag{7.1}$$

This is a smooth rank-one projector with trace one. A local lift is

$$w_\theta = (\cos(\theta/2), \sin(\theta/2)).$$

The formula for Q_θ is periodic, even though the displayed lift reverses sign after one circuit. The associated observational family is $(\theta, x) \mapsto x^{\mathsf{T}} Q_\theta x$.

Theorem 7.1 (Optimal predictor without a global weight)**.** *Let the task at θ have the three fixed inputs above and labels $y_i(\theta) = x_i^{\mathsf{T}} Q_\theta x_i$. For the loss*

$$L_\theta(w) = \frac{1}{2}\sum_{i=1}^{3}\left((w^{\mathsf{T}}x_i)^2 - y_i(\theta)\right)^2$$

the family of global minimizers defines an object in the quotient stationary stack over the parameter circle. Its predictor family is smooth. There is no continuous global weight $w : S^1 \to \mathbb{R}^2$ minimizing all these risks.

Proof. Local half-angle weights make every residual zero, so the minimum is zero. Conversely zero risk implies the equality of the three quadratic evaluations and hence $ww^{\mathsf{T}} = Q_\theta$. Taking traces gives $\|w\| = 1$. With $w = (\cos\phi, \sin\phi)$ the equation becomes $2\phi = \theta$ modulo 2π. The family of local solutions is the degree-two cover of the circle, with the sign action. It defines a principal sign bundle together with the equivariant map to the weight sphere. All local minimizers are stationary, so this is an object of the stationary stack. A global continuous lift would be a section of the doubling map $S^1 \to S^1$. On fundamental groups its composition with the doubling map would give $2\deg(w) = 1$, impossible for an integer degree. The predictor family is the smooth expression in Q_θ given above. □

The loss family uses a fixed architecture and three examples. The obstruction is not caused by a change in the data type or insufficient model capacity. Each task is solved exactly, and any contractible subinterval admits smooth weights. One can use a finite open cover by arcs and read the sign transitions as a nonzero class in $H^1(S^1; \mathbb{Z}/2)$. The two-sheeted cover is the orientation cover of the associated line of principal directions. A display can draw both lifts and show the interchange after one circuit, which is the geometry of the third Rust scene.

7.3 What changes at the fixed point

Adjoin the origin and keep the same sign action. Over the zero predictor there is one weight orbit, but the fiber groupoid at the test point is the classifying groupoid of $\mathbb{Z}/2$. Over a general test and the constant zero predictor, its objects include all principal sign bundles with the constant equivariant map to zero. In particular the zero predictor family need not specify the associated torsor. Smooth evaluation takes it to the same zero function regardless of that choice.

Stationarity has a different fiber structure from optimality here. The gradient of every squared-neuron prediction vanishes at $w = 0$, so $w = 0$ is stationary even for the nonzero label family in Theorem 7.1. It has positive risk for those tasks. The quotient stationary stack includes it and its stabilizer, while the selected global-minimum

subfamily excludes it. Thus taking a stationary fiber and taking a set of optimizers are distinct operations even before one passes to higher test objects.

7.4 The overlap signs explicitly

Take two open arcs of the base circle, one using angles in $(-\pi, \pi)$ and one using angles in $(0, 2\pi)$. Their overlap has two components. On the component with angles in $(0, \pi)$ the half-angle formulas agree. On the other component the two chosen angles differ by 2π, so the local weight vectors differ by a sign. The transition function is therefore $+1$ on one overlap component and -1 on the other. A change of local section on either connected arc can only multiply both overlap signs by the same factor. The two values cannot both be made $+1$. This recovers the nontrivial bundle directly from descent data.

The quotient object is obtained by gluing using those two transition values. Its existence does not require a global choice of gauge, and all local risk and evaluation formulas agree across the two changes of chart. Passing to the observable matrix replaces both weights by the same matrix pointwise. The overlap computation shows precisely which information is forgotten by that replacement.

8 A degree-two obstruction from ineffective symmetry

8.1 From observable families to lifts of their gauge data

An effective symmetry describes actual changes of weights. A presentation can retain a larger group of allowed changes of internal labels whose action on weights has a kernel. This adds automorphisms even where the effective action is free. The following result shows that lifting a predictor family to the larger presentation can have an obstruction in degree two. The construction is a standard central-extension lifting problem applied to the training family, with the distinction between effective and ineffective quotients kept explicit [38, 39, 36].

Let

$$1 \longrightarrow A \longrightarrow \widetilde{G} \xrightarrow{\pi} G \longrightarrow 1 \tag{8.1}$$

be a central extension of discrete groups. Assume that G acts freely and properly on a smooth parameter manifold P, and that $\widetilde{G}$ acts through π. Put $Q = P/G$. With the usual right principal convention let $E_P \to Q$ be the quotient bundle. For a smooth map $q : T \to Q$ write $E_q = q^* E_P$. A point of $[P/\widetilde{G}](T)$ lying over q is equivalently a principal $\widetilde{G}$-bundle $\widetilde{E} \to T$ with an identification $\widetilde{E}/A \simeq E_q$. Indeed the equivariant map to P factors through $\widetilde{E}/A$ and identifies the latter with the pullback bundle. Conversely such a quotient identification recovers the required map to P.

In the smooth ∞-topos the same comparison is the homotopy pullback

$$[P/\widetilde{G}] \simeq Q \times^h_{BG} B\widetilde{G}. \tag{8.2}$$

Here $Q \to BG$ classifies E_P. The classifying map of the extension $BG \to B^2\underline{A}$ makes $B\widetilde{G}$ its homotopy fiber over the trivial class. For discrete groups these are deloopings of the corresponding locally constant group sheaves. Thus the lift is a trivialization of an A-gerbe on the test object. The appearance of $B^2\underline{A}$ has a concrete purpose: it distinguishes different ways local presentations can fail to compose on triple overlaps.

8.2 The obstruction calculation

Theorem 8.1 (Coherent lift of a predictor family)**.** *The object $q : T \to Q$ lifts to the ineffective quotient in (8.2) if and only if a canonical class in $H^2(T; A)$ vanishes. In the nonempty case, the components of the groupoid of lifts are a torsor for $H^1(T; A)$ and the automorphism group of each lift is $H^0(T; A)$.*

Proof. Use a good cover on which E_q is trivial. Write its transition functions g_{ij} and choose lifts $\widetilde{g}_{ij}$. The product

$$a_{ijk} = \widetilde{g}_{ij}\widetilde{g}_{jk}\widetilde{g}_{ik}^{-1}$$

belongs to A. Associativity on a quadruple intersection, together with centrality of A, is exactly the Cech two-cocycle law. Changing the lifts multiplies this cocycle by a coboundary. Changing the local trivializations of E_q conjugates by chosen lifts of the gauge transformations at the relevant endpoints and leaves the class unchanged. If the class vanishes, a one-cochain in A corrects the lifts so they obey the ordinary transition law. Conversely an existing bundle supplies lifts with trivial defect. After one lift has been chosen, any other with the fixed quotient identification differs locally by an A-valued one-cocycle. Isomorphisms differ by A-valued zero-cochains. This gives the torsor and automorphism statements. □

In contrast to the weight-lifting problem in Section 7, we can keep a nontrivial effective principal bundle throughout. What fails here is the existence of a lift of that bundle to the extended group. Locally the ineffective quotient has objects and their automorphisms are copies of A. The obstruction arises from their compatibility around the cover.

8.3 A projective family of exact fits

Use unit weights $P = S^2 \subset \mathbb{R}^3$ and predictor $f_w(x) = (w^\mathsf{T} x)^2$. The effective sign group is $\mathbb{Z}/2$. Its quotient is $Q = \mathbb{RP}^2$, and the predictor associated to $[w]$ is determined by $Q_w = ww^\mathsf{T}$. For each projective point choose the target labels given by its quadratic values at the three coordinate vectors and the three normalized pairwise sums. These six values determine the symmetric matrix. The least-squares risk is zero exactly at the two unit weights over that point. This gives the effective quotient an entire family of optimal predictors over $T = \mathbb{RP}^2$, represented by the identity $T \to Q$.

Now take $\widetilde{G} = \mathbb{Z}/4$ with action through reduction modulo two, so $A = \mathbb{Z}/2$. The pulled-back effective principal bundle is the antipodal cover $S^2 \to \mathbb{RP}^2$. A $\mathbb{Z}/4$ lift of it would give a monodromy homomorphism $\pi_1(\mathbb{RP}^2) = \mathbb{Z}/2 \to \mathbb{Z}/4$ whose reduction is the nontrivial map to $\mathbb{Z}/2$. Every such homomorphism has image in the elements of order at most two, which reduce to zero. Hence the lift does not exist [40]. Equivalently the Bockstein for (8.1) takes the class of the antipodal cover to the nonzero class in $H^2(\mathbb{RP}^2; \mathbb{Z}/2)$. The degree-two obstruction is therefore nontrivial even though the predictor and its optimal effective quotient model exist globally.

The larger symmetry in this example is a specified additional presentation. The predictions and loss do not force that extension to be used. Its effect is on the coherence of lifts, which is why it cannot be detected by the loss values alone. The Rust scene shows the antipodal paths in the weight sphere and the twice-traversed projective loop whose lift is contractible. The order-two relation of the base loop is the relation tested by monodromy into the larger group.

9 Derived critical loci and singular losses

9.1 Two interpretations of a zero locus

The zero locus in Section 6 is formed in sheaves on ordinary smooth test manifolds. At each test, vanishing is an ordinary equality of covectors on the given family. Gauge isomorphisms can survive this operation. Relations between equation residuals are not separate infinitesimal directions in that site. A derived test category provides a second construction.

For a precise algebraic model work over $k = \mathbb{R}$ and use connective commutative derived k-algebras, presented in characteristic zero by commutative differential graded algebras in nonpositive cohomological degrees. For a polynomial loss $L \in k[p_1, \dots, p_m]$ write $A = k[p_1, \dots, p_m]$ and $B = A[u_1, \dots, u_m]$. The two maps $B \rightrightarrows A$ send u_i to $\partial_i L$ and 0, respectively. The derived critical locus is the derived affine with algebra

$$A \otimes^{\mathbf{L}}_{B} A. \tag{9.1}$$

It is the derived intersection of the differential graph and zero section inside the cotangent bundle. One can sheafify derived affines in a chosen derived topology and use their mapping spaces. The local algebra calculation here uses no choice of global compactification or derived smooth enhancement [41, 42, 43].

9.2 Koszul model and the tangent complex

The variables u_i form a regular sequence in B. Resolve the zero section by the free Koszul algebra, then base change along the other map. This represents (9.1) by

$$K_L = \big(A \otimes \Lambda(\epsilon_1, \dots, \epsilon_m), \quad |\epsilon_i| = -1, \quad d\epsilon_i = \partial_i L\big).$$

In degree zero its cohomology is the usual Jacobian algebra $A/(\partial_1 L, \dots, \partial_m L)$. Negative degrees can record relations between the residuals. This construction is coordinate-independent under algebraic changes of coordinates by the cotangent bundle description, which also specifies the transition of the generators. The displayed coordinates provide one convenient free model.

At an ordinary critical point p, the tangent complex is represented in degrees zero and one by

$$T_p P \xrightarrow{\operatorname{Hess}_p L} T_p^* P. \tag{9.2}$$

This can be seen directly by taking the linearization of the two-section intersection, or by dualizing the cotangent complex of the free Koszul model at p. The differential is the derivative of dL at its zero. Thus H^0 is the kernel of the Hessian and H^1 its cokernel. The Hessian at a critical point is an intrinsic symmetric map even though

second derivatives at a general point require a connection to be viewed in this way. If the Hessian is invertible, both groups vanish. The full derived model still depends on the equation beyond its first linearization.

9.3 A quartic loss with nontrivial negative cohomology

Take

$$L(x,y)=\frac{1}{4}(x^2+y^2)^2, \qquad q=x^2+y^2.$$

Its only real critical point is the origin. The two residuals are xq and yq. Its Koszul complex is

$$K^0=A, \qquad K^{-1}=A\epsilon_x\oplus A\epsilon_y, \qquad K^{-2}=A\epsilon_x\epsilon_y.$$

A degree -1 cycle has coefficients (a,b) satisfying $q(ax+by)=0$. Since A is a domain, this reduces to $ax+by=0$. The variables x,y are coprime, so every such pair is a multiple of $(y,-x)$. The image of the differential from degree -2 is generated by $q(-y,x)$. The same domain argument shows that this differential is injective. Consequently

$$H^{-1}(K_L)\simeq A/(q), \quad H^0(K_L)\simeq A/(xq,yq), \quad H^{-2}(K_L)=0. \tag{9.3}$$

The generator of H^{-1} is the class of $y\epsilon_x-x\epsilon_y$. At the real origin the Hessian is zero and both tangent cohomology groups in (9.2) are two-dimensional. The ordinary critical set contains one real point. The derived equation records a nontrivial syzygy and infinitesimal directions there. Algebraically the zero locus over other fields or general rings is different from the real set. In particular one should not infer that the Jacobian algebra is a field from the uniqueness of a real critical point.

For comparison a positive definite quadratic loss in two variables has an invertible Hessian and a Koszul resolution of the single critical point. Its negative cohomology vanishes. The difference between these models is invisible if one records only the set of real critical points. The quartic scene draws the flatness of the loss near the origin. The syzygy itself is established by (9.3), independently of the displayed surface.

9.4 Gauge equivalence and derived equations

If a finite group acts algebraically on the affine parameter space and preserves a polynomial loss, its cotangent action preserves the differential and zero sections. It therefore acts on the derived critical locus by functoriality of the pullback. One can form the quotient in derived stacks. This combines two kinds of information: ordinary gauge arrows from the group action and derived infinitesimal data from the equation. For a continuous group a derived quotient also requires specifying the site and the

group object used there. The smooth quotient theorem in Section 6 does not by itself provide that derived enhancement.

Evaluating the derived affine on an ordinary test algebra through H^0 recovers the ordinary equations: maps from a connective derived algebra into a discrete algebra factor through its degree-zero truncation. A smooth test T can be used in this comparison by evaluating polynomial coordinates in $C^\infty(T)$. This checks the polynomial residuals on smooth families but does not retain the higher directions carried by derived tests. The usual smooth site and the derived site therefore answer different questions. The former is enough for the topological weight-lifting obstruction, and the latter is needed for (9.3) as a derived zero-locus statement.

10 Stochastic learning in Markov categories

10.1 Kernels and tensor composition

For standard Borel spaces X, Y, a Markov kernel $K : X \rightsquigarrow Y$ assigns a probability law $K(x, -)$ to each x, measurably in x. Composition with $L : Y \rightsquigarrow Z$ is

$$(L \circ K)(x, A) = \int_Y L(y, A)\, K(x, dy).$$

The identity is a Dirac kernel. Products of spaces and independent products of kernels give a symmetric monoidal category. Deterministic maps embed by Dirac kernels and preserve ordinary Cartesian copying and discarding. The resulting framework is a concrete Markov category [47, 46, 45].

Copying a sampled variable and sampling twice are different operations. For a fair bit, copying one draw gives mass one-half on each of $(0, 0)$ and $(1, 1)$. Sampling twice gives mass one-quarter on all four pairs. In a wiring diagram the two readings differ by whether copying occurs before or after the stochastic arrow. Deterministic maps preserve copying. A general kernel does not. This distinction matters for shared dropout masks, correlated evaluation experiments, and ensembles reusing a source of randomness.

The smooth topos of Section 5 is not being identified with the category of standard Borel kernels. A smooth deterministic map is measurable and hence gives a kernel, but the internal smooth function space and a measured function type require different constructions. Quasi-Borel spaces offer a model for higher-order probabilistic functions [51]. We use ordinary kernels wherever the underlying measurable space is explicit.

10.2 Conditioning and Bayesian inversion

Given a prior μ on X and a kernel $K : X \rightsquigarrow Y$, their composite defines a joint law on $X \times Y$. Under the standard Borel hypotheses, a regular conditional law $K^\dagger_\mu : Y \rightsquigarrow X$ can be chosen so that

$$\mu(dx)K(x, dy) = (\mu K)(dy)K^\dagger_\mu(y, dx)$$

as joint measures. It is unique up to (μK)-almost sure equality. It depends on the prior and is not generally an inverse under composition of kernels. Composition of Bayesian inverses uses the intermediate marginals and keeps the almost-sure convention. Cho and Jacobs and Fritz develop the categorical structures for these operations [49, 47].

A reverse differential map is different from a Bayesian inverse. A reverse derivative pulls back a covector at a given evaluation point. Bayesian inversion sends an observation

to a law of latent variables given a prior and a joint distribution. These operations can both appear in a learning pipeline. One needs further hypotheses to compare a differential update to an inference update, even if both are depicted with backwards arrows.

10.3 Expectation and differentiation

Let a random sample Z have law μ independent of p, and let $L(p, z)$ be differentiable in p. Suppose on a neighborhood the coordinate derivatives are dominated by an integrable function and the risk is finite at the point of interest. Along a sufficiently short coordinate segment, the difference quotients are bounded by the same function. Dominated convergence gives

$$d\,\mathbb{E}_\mu L(p, Z) = \mathbb{E}_\mu\, d_p L(p, Z). \tag{10.1}$$

The sample reverse derivative is then an unbiased estimator of the risk covector. Sampling strategies that depend on p require care. If the law has a differentiable positive density q_p relative to a fixed dominating measure and the required domination holds, the derivative has the additional score contribution

$$d\,\mathbb{E}_{q_p} L = \mathbb{E}_{q_p}\left[d_p L + L\, d_p \log q_p\right].$$

The equality follows by differentiating Lq_p. Policy-gradient and score-function estimators are among the applications of this identity [58, 59]. Parameter-dependent supports or nondifferentiable densities need separate analysis.

If the random state is supplied by a smooth reparametrization $Z = \phi(p, \xi)$ of fixed base noise, the derivative may instead pass through ϕ under the corresponding domination. The score and pathwise formulas have different inputs and variance behavior. A Markov category identifies the resulting kernel or law. Choosing one representation of sampling noise and a differentiable evaluator supplies additional computational structure.

10.4 Stochastic optimizers and latent state

A stochastic update is a kernel from a state space to itself, possibly indexed by data. One can include optimizer moments, counters, or cached randomness as part of its state. Composition of steps is kernel composition. If the kernels commute with a group action, the corresponding orbit observation is consistent under representative changes when the measured quotient has the required properties. For example a finite group acting measurably on a standard Borel space has a well-behaved Borel orbit space. Smooth quotient-stack descent and measured orbit semantics still retain different data.

Diffusion models provide a related typed example. A finite denoising schedule is a composite of learned reverse-time transition kernels acting on a noise prior. Its training objectives and the particular approximation to time reversal are separate from the Markov composition law [57]. Finite compositions require no assertion that a chosen continuous-time limit exists. This pattern will reappear when separating a language model's token kernel from the surrounding agent loop.

10.5 Couplings and shared evaluation

A comparison of two stochastic predictors on the same input can require a coupling of their randomness. For example a variance estimate for a difference of predictions depends on the joint distribution of the two outputs. The marginal kernels determine the expectations of the separate predictions. They do not determine the covariance term in that variance. A shared random seed supplies a coupling by sampling the seed once and applying both prediction maps. Two independent seeds instead use the tensor product of the sampling kernels. Both are compatible with the same marginal behavior.

The same issue appears in comparing repeated policy rollouts or multiple calls to a stochastic tool. An agent state can carry a common latent random value that influences each call. Treating the calls as independent after forgetting that state changes the history law. The serial composition in Section 12 keeps the joint output and state kernel, which is enough to retain such dependence.

11 Attention, sequence models, and causal sampling

11.1 Attention as a parameterized layer

Take query, key, and value matrices Q, K, V with row counts m, n, n. For fixed scaling $c > 0$ put

$$S = cQK^{\mathsf{T}}, \quad A = \mathrm{softmax}_{\mathrm{row}}(S), \quad \mathrm{Att}(Q, K, V) = AV.$$

In the usual Transformer the three matrices are obtained by learned linear maps applied to input token representations, and c incorporates the key dimension [53]. The layer is a smooth map of finite real arrays. A row of A can be viewed as a probability distribution over key indices. Multiplying by V computes its expected value. This yields a precise relationship to a finite Markov kernel. The kernel itself depends on the query and key arrays. Attention is therefore a data-dependent parameterized operation, even when the value aggregation is linear in V for fixed Q, K.

11.2 Permutation equivariance

Let P_m permute query rows and P_n permute the common key and value rows. If a mask M is added to scores, allow entries $-\infty$ for forbidden pairs with at least one allowed entry in each row. The mask must transform along with the row and column labels.

Proposition 11.1 (Equivariance with an explicit mask)**.** *Under the simultaneous substitutions*

$$(Q, K, V, M) \longmapsto (P_m Q, P_n K, P_n V, P_m M P_n^{\mathsf{T}}),$$

masked attention transforms by left multiplication by P_m. The unmasked case follows by taking $M = 0$.

Proof. The score matrix with its mask transforms by $S + M \mapsto P_m(S + M)P_n^{\mathsf{T}}$. Row softmax commutes with the row reordering, and each row softmax commutes with a permutation of its entries. The new stochastic matrix is $P_m A P_n^{\mathsf{T}}$. Its product with $P_n V$ is $P_m AV$. □

For self-attention without position information, the simultaneous permutation is the same on query, key, and value rows. Shared rowwise maps preserve the same equivariance. Sum or mean pooling can convert it to invariance of an output over all rows. This is related to the design of set architectures [54, 55, 56]. A fixed causal mask breaks the unrestricted permutation symmetry because it records a specific order. Absolute position embeddings have a similar effect when position labels are held fixed. The equivariance statement for arbitrary permutations then requires transforming those labels or using the subgroup that preserves the added data.

11.3 The differentiation of an attention kernel

For an allowed row of scores s with probability vector a, the Jacobian is $\operatorname{diag}(a) - aa^{\mathsf{T}}$ on the allowed entries. Holding the mask fixed,

$$\delta(AV) = (\delta A)V + A(\delta V), \qquad \delta S = c\big((\delta Q)K^{\mathsf{T}} + Q(\delta K)^{\mathsf{T}}\big).$$

The derivative of the weights depends on both query and key variations. Treating attention as a fixed kernel in the value pass is valid only when these inputs are fixed. Reverse mode routes three different cotangent contributions through the layer. The row Jacobian kills constant shifts in the allowed scores. That observation concerns a local redundancy in score coordinates. It does not assert that every such shift is realizable by an invertible weight transformation of the original architecture.

11.4 An autoregressive model and a sampling policy

For a finite token alphabet $\mathcal{V}$, a language model can be evaluated on a prefix to give a probability distribution of the next token. Writing that kernel as $K_\theta(v \mid h)$, the law of a finite continuation is the product of the conditional probabilities evaluated at successive prefixes. The same weight θ is used at each evaluation, even if the state representation caches intermediate data. A sampling policy can use temperature, select a maximum, truncate candidates, or call a constrained decoder. Those operations modify the transition kernel. Greedy decoding is deterministic after a tie-breaking rule and can be discontinuous in the scores.

The training graph and the deployed state machine may therefore have different interfaces and regularity. For example token-level cross-entropy differentiates through a predictive distribution given a fixed prefix. Differentiating a loss of sampled future trajectories requires either a stochastic estimator or an additional differentiable representation. The algebra of parameterized layers provides the single-evaluation semantics. The operational semantics of agent loops will make the environment and the history of observations explicit.

12 Open stochastic agents and temporal composition

12.1 A finite-step agent interface

An agent with input X and output Y is described by a standard Borel state S, initial law μ on S, and a transition kernel

$$K : S \times X \rightsquigarrow S \times Y. \tag{12.1}$$

State can contain a context, a tool transcript, a working policy, or a learned parameter vector. An output might be a token, a control action, or a typed request to an external system. On this interface a single step returns both an updated state and an output. In a closed environment the input comes from an environment transition or from an external input protocol. The typing records which data flow between those components.

Suppose K is followed in the same time step by $L : T \times Y \rightsquigarrow T \times Z$. Use joint state $S \times T$, sample (s', y) using K and then (t', z) using L with that same y. The composite kernel on a test set $A \subseteq S \times T \times Z$ is

$$\int_{S\times Y} \int_{T\times Z} \mathbf{1}_A(s', t', z)\, L(t, y, d(t', z))\, K(s, x, d(s', y)).$$

If initial states are chosen independently the initial law is the product. A correlated initial law can also be supplied explicitly at the composite interface. It encodes additional information that is absent from the independent composition of the two pointed agents.

Proposition 12.1 (Serial composition)**.** *For standard Borel interfaces and states, the preceding operation is associative up to the canonical product associator on joint state. A deterministic wire with singleton state is an identity. Independent parallel composition uses products of kernels.*

Proof. Three serial agents give the same nested integral over the three successive sampled pairs for both bracketings, after relabeling the product state. The equality follows from composition of kernels and Tonelli's theorem on nonnegative tests. A Dirac kernel on an unchanged input inserts no sampling choice and leaves the composite kernel unchanged. The product of independent kernels implements parallel steps by definition. □

There is a corresponding deterministic construction by replacing each kernel by a map. It belongs to the broader study of coalgebras, open dynamical systems, and compositional games [61, 63, 62, 65]. A finite agent can be unrolled into a causal diagram. The same transition and state are used at every step, so the unrolling preserves dependence between successive outputs.

12.2 Delayed feedback

Suppose a step consumes (x, a) and outputs (y, a') in addition to a new state. Store a in a register and use a' as the register value for the next step. With a specified initial value or law for the register, this defines a transition on $S \times A$. At every finite horizon the register makes the diagram causal and each kernel composition well defined. Wiring the output directly back into the same step can instead require solving an implicit equation or specifying a fixed-point selection. The delayed model supplies an explicit operational meaning and does not require a trace operator for arbitrary instant cycles.

For instance a tool response can be placed in the register and consumed at the next model call. The tool may itself have state and a failure distribution. A tool invocation and its reply then have different types and occur in a specified temporal order. Changing the order can change the distribution even if the same stateless components appear in the program.

12.3 Simulations preserve finite observations

Let $h : S \to T$ be a measurable map between two agents on the same interfaces. Suppose $h_*\mu = \nu$ and

$$(h \times \mathrm{id}_Y)_* K(s, x) = L(h(s), x) \tag{12.2}$$

for every s, x. This is a deterministic state simulation with equality of the resulting kernels.

Theorem 12.2 (Finite-trace preservation)**.** *Under* (12.2)*, the output-history laws agree at every finite horizon under any common causal input protocol that uses only past external inputs and outputs.*

Proof. At time zero pushforward of the initial state law gives the claimed identification. Suppose the joint law of state and external history of the second agent is the pushforward of the corresponding joint law of the first. The common protocol samples the same input law conditional on that history. Equation (12.2) identifies the new joint laws after applying each transition and pushing the first state forward along h. Induction gives the statement. Marginalizing state leaves the equal output-history laws. □

Trace agreement alone does not determine an isomorphism of state machines. A hidden bit can alternate or stay fixed while the output is always zero. These two deterministic transition systems have the same external trace laws for any input protocol, but a state bijection cannot conjugate the transposition to the identity. If one allows a quotient that forgets the bit, both admit a simulation to a singleton-state constant agent. The observation criterion should be fixed before drawing conclusions about internal mechanisms.

12.4 Path laws and feedback from learning

Consistent finite histories on standard Borel interfaces have the familiar extension to a law of infinite histories when the required sequence of kernels and initial law are specified. This is a measure-theoretic statement about countable products. A formal limit of smooth predictor stacks does not supply it. Categorical treatments of infinite stochastic products give a separate account of this extension principle [52].

Learning in the loop can be included by carrying weights and optimizer state in S and applying an update based on the received observation. Freezing the learned weights gives a different agent, even if the two agree for their first outputs. Reinforcement learning compares policies by rewards and transitions, not just supervised prediction error. Partially observable models distinguish environmental state from what the policy can access [59, 60]. The interface (12.1) can express both cases by an appropriate state and kernel. A correctness claim about a particular agent still depends on its concrete policy, environment, and stopping convention.

13 Federated learning, descent, and the arithmetic of updates

13.1 Data descent and finite sums

Represent a dataset by a finite set of record identifiers, each with a sample and label. A cover by subsets with agreed values on overlaps glues to a unique dataset on their union. This is a sheaf condition for assignments on a fixed underlying set of record identifiers. If the identifiers or their matching are part of the problem, the cover must also supply the identifications. Identical feature values alone do not distinguish a duplicated transmission from two distinct observations.

For a fixed p define the unnormalized risk on a finite subset U by $R_U(p) = \sum_{i \in U} \ell_i(p)$. On disjoint subsets these risks add. For two overlapping subsets,

$$R_{U \cup V} = R_U + R_V - R_{U \cap V},$$

and the same identity holds for their differentials. This follows from the corresponding inclusion-exclusion identity for sample multiplicities. The normalized risk also requires knowing the relevant counts. It does not generally glue by unweighted averaging of the normalized local risks. Losing record identifiers or their overlap information can change the resulting global objective.

13.2 One synchronized step

For a disjoint partition with client counts n_i and $N = \sum_i n_i$, let $\lambda_i = n_i/N$ and L_i be the mean client risk. Then the global mean is $L = \sum_i \lambda_i L_i$. At a common parameter p,

$$dL(p) = \sum_i \lambda_i \, dL_i(p). \tag{13.1}$$

On a Euclidean parameter space a weighted average of one client step $p_i = p - \eta \nabla L_i(p)$ is exactly $p - \eta \nabla L(p)$. This uses equal step sizes, a common starting parameter, common metric, and one full client gradient in the objective as defined. It does not use similarity of the client risks. One can state the cotangent averaging identity on a manifold because all covectors are at the same point. Averaging displaced parameters on a manifold requires a separate geometric operation.

FedAvg and related methods use several local steps to reduce communication [67, 68]. Once clients move away from the common point, their next gradients are evaluated at different locations. This is the source of a simple exact comparison for quadratic losses.

13.3 An exact two-step discrepancy

Let L_i be smooth quadratic functions on a common Euclidean space, with constant Hessians H_i. Set $g_i = \nabla L_i(p)$, $\bar{g} = \sum_i \lambda_i g_i$, and $\bar{H} = \sum_i \lambda_i H_i$. The second client gradient is $g_i - \eta H_i g_i$. The average of two-step local updates is therefore

$$p - 2\eta\bar{g} + \eta^2 \sum_i \lambda_i H_i g_i.$$

Two synchronized centralized steps give $p - 2\eta\bar{g} + \eta^2 \bar{H}\bar{g}$. Their difference is

$$\eta^2 \left(\sum_i \lambda_i H_i g_i - \bar{H}\bar{g} \right). \tag{13.2}$$

For two equally weighted clients the parentheses equal $\frac{1}{4}(H_1 - H_2)(g_1 - g_2)$. The condition for equality is thus a vanishing mixed term. Different local gradients can give the same result when the Hessians agree, and different Hessians can give the same result when the gradients agree at the starting point.

For the scalar risks

$$L_1(p) = \tfrac{1}{2}(p-1)^2, \qquad L_2(p) = \tfrac{3}{2}(p+1)^2,$$

start at $p = 0$ and use $\eta = 1/4$. The first averaged and centralized step is $-1/4$. After two local steps the client parameters are $7/16$ and $-15/16$, with average $-1/4$. The centralized second step reaches $-3/8$. The difference is $1/8$, as (13.2) also gives. The fourth comparison of the viewer draws the risks and the competing iterates in a common spatial parameter chart.

13.4 Why averaging and quotienting can disagree

If the same predictor has several parameter representatives, client endpoints cannot generally be averaged independently of their representatives. In the squared-neuron model let two clients return w and $-w$ for a fixed nonzero w. Both give the same predictor. Their average is the zero vector, which gives the zero predictor. Choosing a common representative would instead give a nonzero average. The failure occurs even before any stochastic gradient noise.

This shows that addition of weight vectors is not a function of independent parameter orbits. One can first align representatives using additional information, compare predictions directly, or operate on an explicit quotient representation. Each choice gives its own aggregation rule. Permutation-based model alignment is studied in [31, 32]. The higher-topos formulation separates three operations: gluing local presentations with overlap symmetries, adding contributions to a risk at a common parameter, and averaging displaced points in a chosen coordinate system. The first is descent, the second is additive, and the third depends on the geometry of the parameter representation.

14 Observation, expressivity, and composition across levels

14.1 The observable image of a model stack

For an architecture quotient $\mathcal{M}$ and evaluation $e : \mathcal{M} \to \mathcal{F}$, factor e into its effective epimorphism onto the image followed by a monomorphism into $\mathcal{F}$. In the smooth topos the image is the subobject of predictor families that are locally realized by objects of $\mathcal{M}$. Because an atlas is itself locally surjective, it is also the sheaf image of the atlas evaluation $P \to \mathcal{F}$. At each point one can instead take the ordinary set image of $P(T) \to \mathcal{F}(T)$. The two notions can differ: the image subsheaf retains families that become realizable after a cover, even if no global atlas parameter exists. Section 7 gives an explicit family for which the ordinary set image of global weights is smaller.

Evaluation may also forget symmetries in the fiber. At a point, an automorphism of a model with fixed predictor becomes the identity on that predictor. Even passing to a sheaf of components can leave distinctions between different orbits that happen to implement the same function. Taking an observable image, passing to components, and choosing a gauge are therefore different categorical operations. They can coincide under additional free-presentation and identifiability hypotheses, which the preceding results state where needed.

14.2 Evaluation on a dataset

For a fixed finite set of inputs $x_1, \dots, x_n$, evaluation on those inputs gives a map $\mathcal{F} \to Y^n$. Two functions with the same image can differ on other inputs. If the objective depends on the parameters only through these evaluations, they receive the same empirical loss. The differential can still differ between their parameter representatives. Equality of loss values at a point does not identify tangent behavior of the programs or their optimizer trajectories.

A separate statistical argument is needed to compare empirical risk and population risk. For example take a finite class $\mathcal{G}$ of predictors and a loss valued in $[0, 1]$ on i.i.d. samples. Hoeffding's inequality and a union bound imply

$$\Pr\big[\sup_{g \in \mathcal{G}} |L_n(g) - L(g)| > \epsilon\big] \le 2|\mathcal{G}|e^{-2n\epsilon^2}.$$

Here L_n is the empirical mean and L the population expectation [71]. This bound can use the number of distinct functions even if many parameters realize each function. The identification is appropriate for this observable bound. It says nothing about gauge stabilizers, parameter conditioning, or runtime cost. For infinite classes one needs another capacity or stability argument.

14.3 Real semantics and implemented finite maps

A fixed finite-precision implementation with fixed input and output encodings and a total bounded computation defines a function between finite bit strings. Any such function is extensionally implementable by a Boolean circuit. For example a disjunction of minterms for the inputs yielding one gives a circuit for each output bit. This argument gives existence and can be exponentially expensive. It does not transport smooth derivatives, state representations, or a uniform resource bound. For a fixed stochastic runtime the selected random input bits can be included in the input to the deterministic implementation, with the induced output law determined by their distribution.

Conversely the smooth model used to differentiate a network describes its intended real operations. Replacing them by an implementation can introduce rounding, overflow, and branching. One can seek a compiler correctness theorem that relates real and machine semantics under stated error hypotheses [21, 19]. The categorical reverse rule in Section 3 is an exact identity in the smooth model. Testing it on selected floating-point runs checks a different property.

14.4 Approximation and architectural semantics

Universal approximation theorems concern density of a function class under specified hypotheses on activation, compact domain, norm, and architecture [69, 70]. They do not construct a given trained model, identify its gauge representatives, or bound the cost of finding it. A compositional description of architecture instead organizes how the maps are assembled. The claim that category theory offers an algebraic theory of architectures is developed explicitly by Gavranović and coauthors [8]. Their structural program and the analytic approximation theorems address different comparisons.

The XOR surface below is an exact low-dimensional example: the polynomial $u+v-2uv$ realizes all four Boolean labels and is smooth on the whole real plane. A real affine score with a threshold cannot strictly separate the two labels at the four corners, because the averages of its scores on the two diagonals are equal. A polynomial feature solves the task, and many other programs solve the same four-point task. Evaluation on those four inputs provides no canonical parameterization or extension outside the square.

14.5 Limits for coherent training constraints

Let a finite diagram of learning components be given in an ∞-topos. Each component has a space of local representations and maps that expose its interface data. A compatibility problem can be defined as the limit of the diagram that matches these interfaces. Its points contain components and coherent identifications. If the stationary

conditions are natural in the same diagram, the limit of componentwise stationary stacks agrees with imposing the assembled equations after the matching. This follows because limits commute with limits. The naturality of the equations is the substantive hypothesis. It may fail for an objective that uses interactions across components, or for a change of architecture that creates additional variation directions.

For example, a shared parameter imposes the diagonal restriction in Section 2. The constrained stationary equation is the sum of the cotangents of its uses. Requiring each use to be stationary separately imposes a stronger condition. Two blocks can have nonzero opposite covectors at a shared parameter. The sum then vanishes. This illustrates why one must specify the differential equation after the interface constraint, and why commutation of limits does not establish naturality of a differential operator.

15 A stack of optimizer histories

15.1 The graph of a training step

Let S be a smooth state manifold carrying weights and any optimizer memory, and let G act on S by changes of presentation. Use a dataset sheaf D and a total equivariant step $U : S \times D \to S$, with trivial action on D. Put $\mathcal{X} = [S/G]$. The step descends to $\overline{U} : \mathcal{X} \times D \to \mathcal{X}$ and defines a span

$$\mathcal{X} \xleftarrow{\mathrm{pr}_1} \mathcal{X} \times D \xrightarrow{\overline{U}} \mathcal{X}.$$

The left leg exposes the state before the step and the right leg exposes the state after the step. Finite histories are formed by iterated homotopy pullback of these spans. A history keeps the sequence of data and the identifications of intermediate states. This is the span of a deterministic operation. A more general span may describe a relation with several possible successors.

Theorem 15.1 (History descent for a total equivariant update)**.** *For each finite n, the stack of n-step histories is equivalent over its initial state and data to $\mathcal{X} \times D^n$. Under that equivalence its final-state map is the iterate of $\overline{U}$ with those n data. A history over a test T can be represented by a single principal G-bundle on T with an equivariant initial-state map. Successive equivariant maps on that same bundle give all later states.*

Proof. For one step the graph description gives the equivalence. For the inductive step let the right leg of the history span be its final-state map F_n. Pulling the next left leg $\mathcal{X} \times D \to \mathcal{X}$ back along F_n is canonically product with D. The new right leg is the composition of F_n and the next data with $\overline{U}$. Induction gives the asserted equivalence and iterate. In terms of principal bundles, $\overline{U}$ applies U to the equivariant state map without changing the bundle. If a different bundle is used in a presentation at an intermediate stage, the pullback comparison identifies it with the previous output bundle. These identifications transport the entire history to the initial bundle. The construction is natural in T and hence satisfies descent. □

This result uses a total deterministic step and records enough state for it to be Markovian in the operational sense. If the optimizer has momentum, forgetting it can make the next weights depend on an earlier step not recoverable from the current predictor. One can change the state type or use a relation on predictors. The latter will generally have more than one candidate successor and no canonical probability law for choosing one. The measured agent construction of Section 12 supplies such a law when a kernel is given.

15.2 Terminal conditions and preserving presentation data

A subset or stack of acceptable final states can be imposed by base change along the final-state map. For example a stopping condition can be the zero-covector equation for a prescribed evaluation dataset. The resulting history object classifies initial states and data whose iterates satisfy that condition. It does not assert that all trajectories reach it or that the condition is equivalent to dynamical convergence. A stationarity test for the weight risk can hold while optimizer memory still leads to a nonzero next update.

Because the bundle is preserved in Theorem 15.1, a sequence of total equivariant updates cannot trivialize the underlying principal bundle carried by an input family. Its evaluations may lose evidence of that bundle. For example an equivariant map to a group-fixed parameter can make the predictor constant while retaining the original bundle in the quotient stack. A rule that chooses a global weight chart for the family would have to address the lifting obstruction independently of the optimization step.

There is a pointwise restriction on stabilizers as well. If $g \in G$ fixes a state s, equivariance gives $gU(s,d) = U(gs,d) = U(s,d)$. Thus the stabilizer of a state is contained in the stabilizer of its image, for the same data. An invertible equivariant map has equality. A deterministic step on a complete state cannot break a symmetry that fixes that state. In actual training, auxiliary state, data, or noise can transform too. Then the symmetry test has to be made on the joint input.

15.3 Histories, limits, and observable transition laws

Taking the inverse limit of finite history stacks classifies compatible finite deterministic histories. If a total update and an infinite data stream are supplied, its sections are described by the initial-state family and the successive data under the same canonical comparison as in Theorem 15.1. The limit in sheaves imposes componentwise smoothness of all finite restrictions. It does not impose a norm topology or uniform bounds on the sequence of states. Those properties can be imposed only by adding a compatible state condition.

In a stochastic model, a support relation can have the same possible finite paths for two different transition laws. For example two Bernoulli kernels with different probabilities strictly between zero and one both allow every finite bit string. Their probabilities for those strings differ. Thus a limit of support relations records possible histories, whereas composing kernels records observable distributions. This distinction also explains why adding probability to the smooth quotient requires a comparison of measurable structures, beyond taking an ordinary homotopy limit in the smooth topos.

The treatment of open learners in categorical cybernetics and recent compositional dynamics work gives broader constructions of adaptive state machines from interfaces and potentials [66, 64]. The result here is a fixed-state-space comparison for a specified update, designed to expose how optimizer histories inherit the descent and bundle information of the model.

15.4 Transporting optimizer histories

Suppose two deterministic training systems have state stacks $\mathcal{X}, \mathcal{Y}$ and share the dataset sheaf D. Let $h : \mathcal{X} \to \mathcal{Y}$ intertwine their steps as morphisms over the data. Then h induces a comparison of the spans defining histories. The map on n-step histories is identified by Theorem 15.1 with $h \times \mathrm{id}_{D^n}$. The two final-state maps commute with this comparison. The proof is either induction on the iterate or functoriality of the iterated pullbacks.

An equivalence of state stacks gives equivalence of the histories if the intertwining condition holds. By contrast an equivalence of predictors can fail to produce an intertwining map, as the metric example in Section 4 demonstrates. Adding a terminal condition preserves the comparison when the same condition is pulled back along the final-state map. Thus the history construction specifies exactly what must be transported to compare a training workflow rather than just its outputs.

16 Six exact geometries rendered in Rust

The companion uses Rust and Macroquad's three-dimensional camera, lines, and mesh primitives [72]. The scenes are generated from the formulas below. They display finite samples of smooth surfaces or exact iterations. The mathematical statements use the formulas and proofs, while the viewer makes the geometry of each comparison inspectable. Each scene can be orbited and zoomed independently.

16.1 Two metrics on one loss bowl

The bowl is $L(x, y) = (x^2 + 4y^2)/2$. Set $\eta = 0.08$ and start at $(x_0, y_0) = (1.34, 1.05)$. The physical Euclidean updates and the updates after transporting that metric through the scaled chart both give

$$(x_k, y_k) = \big((1-\eta)^k x_0, (1-4\eta)^k y_0\big).$$

If the second chart instead uses its own Euclidean metric, the physical coordinates are

$$(\widetilde{x}_k, \widetilde{y}_k) = \big((1-\eta/4)^k x_0, (1-16\eta)^k y_0\big).$$

At the selected step size the second factor in the latter trajectory is negative, so its y coordinate changes sign while decaying. Both paths are shown on the same physical loss surface. The display rescales the vertical direction and lifts points slightly off the mesh so they remain visible. Changing the step count reveals the two update rules from the common start. These exact trajectories implement the comparison of Section 4.

16.2 A smooth exact XOR surface

On the square $[0, 1]^2$ put $f(u, v) = u + v - 2uv$. At the two corners with different input bits the value is one. At the other two corners it is zero. The threshold $f = 1/2$ consists of the two lines $u = 1/2$ and $v = 1/2$, because

$$f(u, v) - \tfrac{1}{2} = -2(u - \tfrac{1}{2})(v - \tfrac{1}{2}).$$

The surface is a saddle. Its geometry explains why a single affine threshold fails on this task and why the product term suffices. A parameter-free Cartesian circuit can build the map from copying, multiplication, and addition. One could also use the polynomial as a member of a three-parameter linear family over features (u, v, uv). The figure shows the exact learned function, not a claim about a particular optimizer finding those coefficients. The corner glyphs encode labels, and the gold crossing traces the half-height threshold.

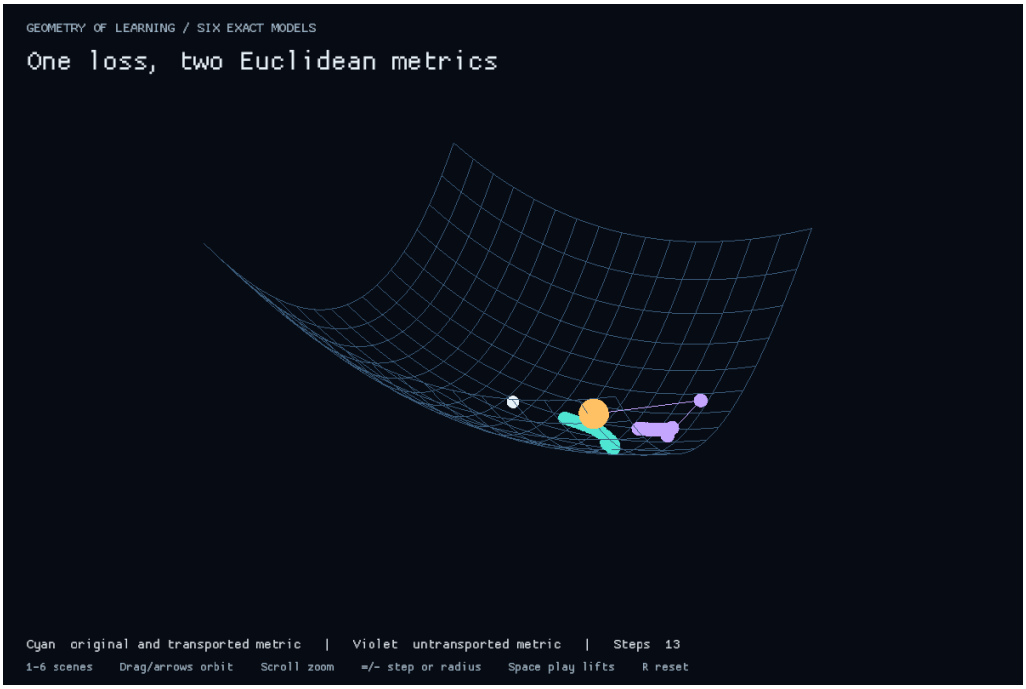


Figure 1: One quadratic risk with two Euclidean descent paths in common coordinates. The cyan path is also obtained with the transported metric.

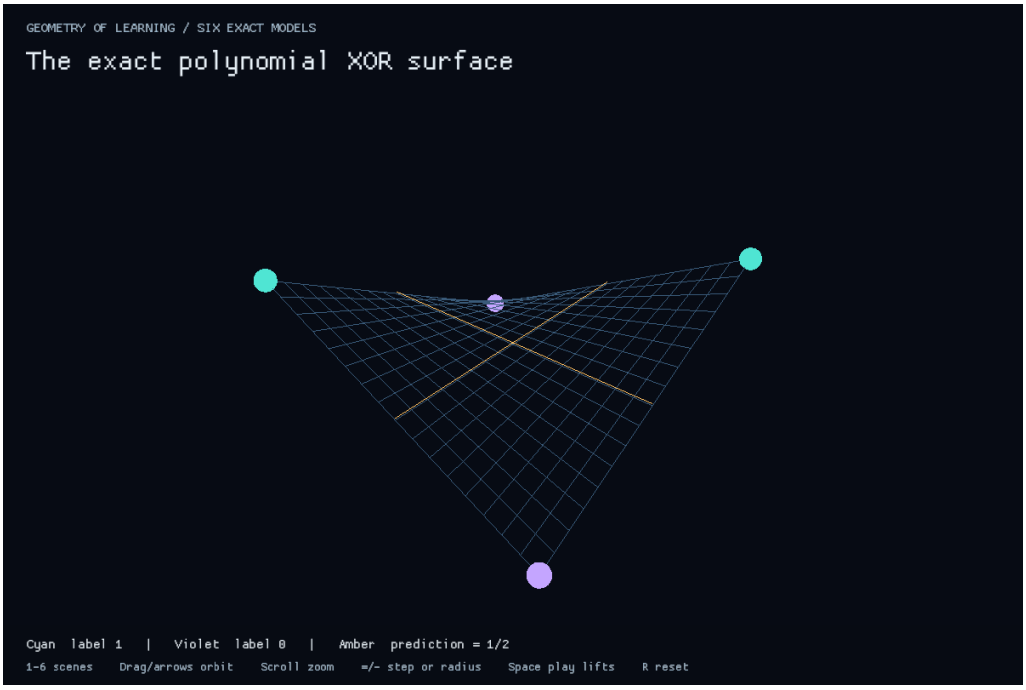


Figure 2: The polynomial XOR surface on the unit square. The two gold curves are the level set of the classification threshold.

16.3 The double cover of optimal predictors

In Theorem 7.1 the unit weight angle ϕ maps to the predictor angle $\theta = 2\phi$. The viewer embeds the total circle in three dimensions using

$$\big((R + a\cos\phi)\cos 2\phi,\ a\sin\phi,\ -(R + a\cos\phi)\sin 2\phi\big), \qquad 0 < a < R.$$

Projection to the gold base ring keeps the angle 2ϕ. The two lifted points over that angle lie on opposite sides of a cross-section. The two colored beads move along this closed curve. After one base circuit they exchange, and after two they return. The short connecting line illustrates the fiber rather than a path within the space of optimal weights. A global continuous weight would select one bead consistently over the entire base. The displayed cover and the degree argument show why the local choices cannot do so.

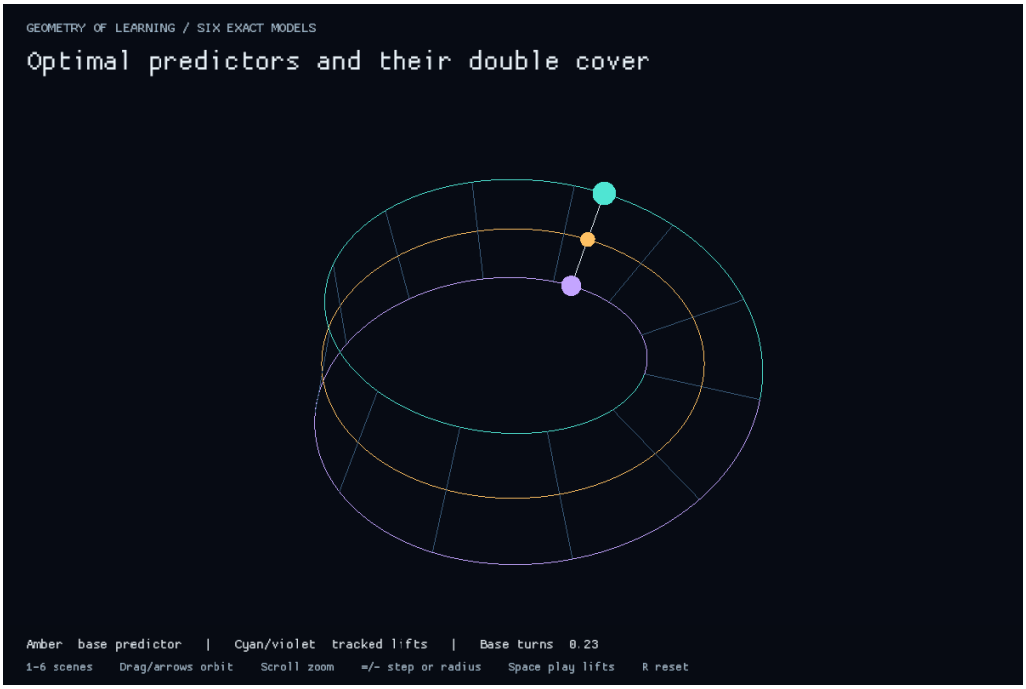


Figure 3: The degree-two cover of the predictor circle. Two lifted weights lie over the highlighted base predictor.

16.4 Federated quadratic trajectories

The fourth scene uses the scalar losses from Section 13. The two client risks and their mean occupy three separate lanes, with the same parameter axis in all lanes. At step two the independent local endpoint average equals $-1/4$ and the centralized endpoint

equals $-3/8$. The viewer places both on the global-risk curve so their parameter values can be compared directly. Moving from zero to one step shows the initial exact agreement. Moving to two steps exposes the mixed Hessian–gradient term in (13.2). At other selected step counts the display compares one round with that many local steps against the same number of centralized steps. It does not insert intermediate client synchronizations.

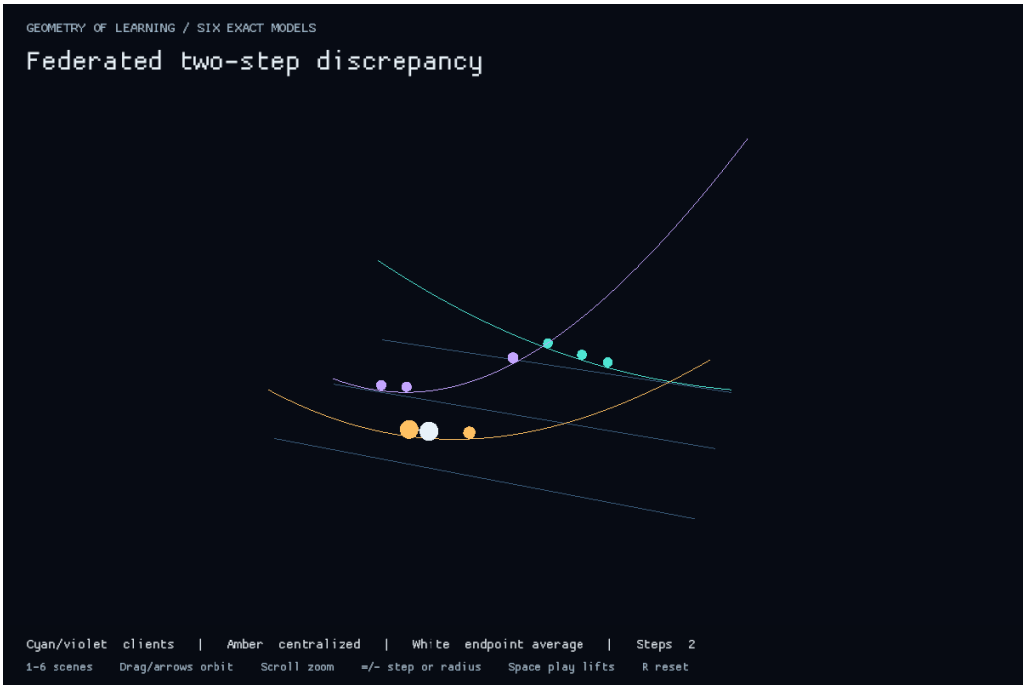


Figure 4: The two client risks and the centralized mean in separate lanes. White and gold mark the competing endpoints on the mean-risk curve.

16.5 The flat bottom of a quartic loss

The fifth scene draws $L(x, y) = (x^2 + y^2)^2/4$. At radius r the height is $r^4/4$, so halving the radius reduces the height by a factor of sixteen. The two level circles represent this radial scaling. A quadratic bowl would reduce height by a factor of four. The central glyph represents the single real critical point. The tangent and obstruction spaces and the syzygy in (9.3) are algebraic features of the derived critical locus. The viewer renders the ordinary surface that supplies the residuals for that calculation.



Figure 5: The quartic loss and the two concentric test radii around its unique real critical point.

16.6 Antipodal monodromy and the projective plane

In Section 8 the unit weight sphere maps to projective predictors by antipodal identification. A generator of the quotient fundamental group is represented by traversing half of the weight equator. The displayed colored semicircles are the two halves of that equator. The two moving beads are opposite weights of the same predictor. After one projective turn the first bead arrives at the initial position of the second. Two turns give a full circle upstairs. That circle contracts in the weight sphere, so its projected loop also contracts in the projective plane. This is the order-two relation tested by the attempted lift of monodromy. Any odd element of $\mathbb{Z}/4$ maps to the required sign change, but its square is nonidentity. The display renders the geometry of that relation. The central-extension calculation identifies the cohomological obstruction.

16.7 Running and reproducing the figures

The package contains the Rust source and a locked dependency graph. From the viewer directory run `cargo run -locked`. Keys 1 through 6 select the scene. Drag or use the arrow keys to orbit. Scroll to zoom. The equals and minus keys adjust step count, radius, or the cover position according to the scene. Space animates the cover and the

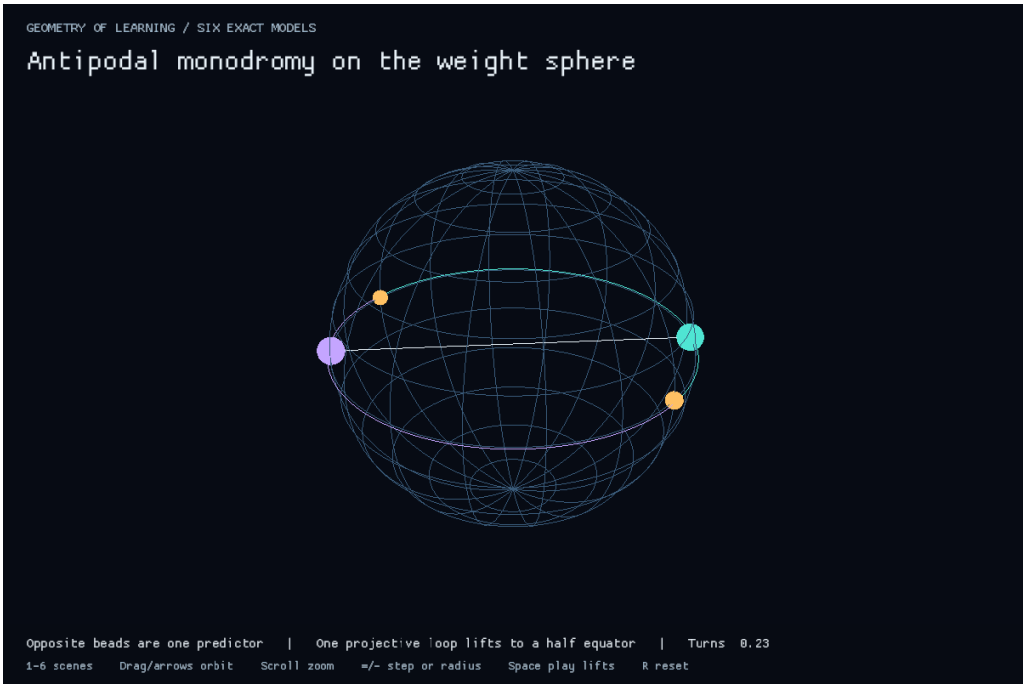


Figure 6: Antipodal weights over one projective predictor. A half equator represents the generator of projective monodromy.

antipodal lift and `R` resets the camera and numerical inputs. To export all six figures run `cargo run -locked - -capture DIR`. The capture path still requires a graphics context. The figures used in this paper were generated by this mode.

17 Notes on the literature and on the scope of the results

17.1 Composition, learning, and differentiation

1. Backpropagation as a functor. Fong, Spivak, and Tuyéras attach a learner to a parameterized function using an error function and a gradient update [1]. The learner contains an implementation, parameter update, and a request passed upstream. Their theorem is stronger than simply saying that derivatives satisfy a chain rule: it identifies composition of the selected learning interfaces. Our basic reverse lens keeps the output cotangent as its backward input. A particular request operation can be added when the error model supplies it. These distinctions matter when optimizers carry state or when the objective is not a per-example scalar loss.

2. Lenses and learners. Fong and Johnson compare the categorical forms of lenses and learners [2]. Asymmetric lenses expose a forward observation and a backward response. More general optics let a residual context survive between those stages. Riley gives an account of optics in monoidal categories [13]. In the concrete smooth example here, the residual information is the forward point needed to evaluate a dual derivative. The groupoid quotient construction concerns parameter symmetries, independently of whether the differentiation layer is expressed with lenses or with a more general optic.

3. Parameterization and architectures. Cruttwell, Gavranović, Ghani, Wilson, and Zanasi combine parameterization, lenses, and reverse differential categories [3]. The categorical deep-learning thesis [5] and the position paper [8] develop architectural and optimization structure in broader settings. Gavranović's early compositional work and the later exposition of parametric lenses give other entry points [6, 7]. Our use of a product of parameter spaces is one concrete actegory situation. We then impose a stationary equation on the moduli of presentations. Architecture changes that are allowed in the parameterized category can fail to preserve that equation because the allowed variation directions have changed.

4. The differential calculus. Cartesian differential categories abstract forward differentiation [14]. Cockett and coauthors give the reverse analogue with a reverse chain rule [4]. The differential lambda calculus studies differential structure at the level of typed program terms [15]. Elliott and Pearlmutter and Siskind give compositional or functional accounts of AD [16, 17]. Abadi and Plotkin specify a differentiable programming language [18]. This manuscript uses the ordinary dual derivative on smooth finite-dimensional maps as its concrete interpretation. A higher-order language

would additionally need a model of function types and its chosen differentiation constructs.

5. Implementation and correctness. The survey of Baydin and coauthors locates forward and reverse AD among the techniques used in machine learning [19]. Huot, Staton, and Vákár study semantic correctness of AD with diffeological tools [20]. Lee and coauthors address machine-representable parameters [21]. Our finite-dimensional calculations describe the mathematical real model. A floating-point implementation has a separate operational semantics. The viewer samples simple formulas and is reproducible from its locked dependency graph. Its output is evidence for the reproducibility of the scenes and is independent of the proofs of the structural theorems.

17.2 Geometry, stacks, and obstruction theory

6. Optimization as a geometric operation. Amari formulates the natural gradient using information geometry [22]. Martens gives a detailed analysis of the method and its relation to practical optimization [23]. Absil, Mahony, and Sepulchre give the general manifold optimization framework [25], and Bonnabel studies stochastic gradient descent on manifolds [26]. Song, Song, and Ermon examine the effect of higher-order discretizations on invariance [24]. The key comparison here is between covector naturality, metric transport, and a displacement rule. All three can be stated on the same parameterized model.

7. Parameter symmetries and flatness. Dinh and coauthors analyze how function-preserving reparametrizations can change parameter-space sharpness [27]. Godfrey and coauthors study symmetries of internal representations [28]. Theiss and coauthors analyze degeneracy of representational geometry at fixed function in overparameterized networks [29]. Kunin and coauthors connect symmetries to constraints on learning dynamics [30]. The stationary stack uses a specified action. It does not assert that this group classifies all equal predictors or that a chosen optimizer preserves the action. Finite permutations and continuous rescalings can have different effects on an optimization method.

8. Weight alignment. Ainsworth, Hayase, and Srinivasa study merging models after finding appropriate permutation alignments [31]. Entezari and coauthors investigate permutation symmetry and mode connectivity [32]. The sign example in Section 13 is a minimal algebraic reason alignment can matter: averaging independent representatives of the same orbit can give different predictor functions. Alignment can be useful without providing a global section of every parameter-presentation bundle.

9. Higher-topos input. Lurie's higher topos theory supplies sheaves of spaces, objectwise limits in presheaves, and effective epimorphism images [33]. Hollander gives a homotopy theory of groupoid stacks [34], and Dugger, Hollander, and Isaksen develop hypercover descent [35]. We need only a small part of that general machinery for the ordinary stationary stack. The local model groupoid is already enough to keep gauge arrows and principal bundle gluing. The ambient topos also supplies the higher classifier used in the lifting problem.

10. Principal bundles and symmetry quotients. Nikolaus, Schreiber, and Stevenson study principal ∞-bundles and their concrete presentations [36, 37]. A smooth principal G-bundle with an equivariant map to a weight space is the ordinary instance of this picture. The nontrivial principal sign bundle in Theorem 7.1 is a family of training solutions because its local maps minimize the same smoothly varying risks. The geometric theorem does not require a numerical training algorithm to choose those weights.

11. Gerbes and central extensions. Giraud develops nonabelian cohomology and gerbes [38]. Brylinski treats gerbes and related classifiers in geometric settings [39]. Our explicit lifting calculation uses the triple-overlap defect of a central extension, so the coefficient group is abelian. The two examples separate a degree-one obstruction to selecting effective weights from a degree-two obstruction to lifting the effective bundle. In the projective example, the monodromy calculation also gives a direct topological test for the nontrivial class [40].

12. Derived zero loci. Vezzosi studies derived critical loci and their basic structure [41]. Pantev, Toën, Vaquié, and Vezzosi place derived intersections in shifted symplectic geometry [42]. Toën's survey gives the broader derived algebraic context [43]. Our computation uses the affine Koszul presentation of a polynomial risk. It identifies negative cohomology explicitly. The paper does not need a general claim about shifted symplectic forms on all models of learning.

13. Smooth and derived smooth models. Spivak's derived smooth manifolds provide a setting in which smooth intersections can be refined [44]. We keep the ordinary smooth site for the global family theorems and the polynomial derived site for the explicit Koszul result. A derived smooth extension would need its own choice of test category and a comparison of critical intersections. The two chosen settings suffice to exhibit the different kinds of information.

17.3 Probability, sequence modeling, and agency

14. Probability as a categorical operation. Lawvere's early account of probabilistic mappings and Giry's probability monad are important antecedents [45, 46]. Fritz develops Markov categories and their synthetic probability theory [47]. Fong gives a categorical account of causal theories and Bayesian networks [48]. The tensor separates independent product from copying. This distinction is needed even for simple learning pipelines that reuse a random value.

15. Conditioning and higher-order probability. Cho and Jacobs formulate disintegration and Bayesian inversion diagrammatically [49]. The existence of a conditional kernel and its almost-sure uniqueness are part of the concrete measured model. Fritz and Perrone study probability measures through colimits of finite samples [50]. Heunen and coauthors give quasi-Borel spaces for higher-order probability [51]. These theories explain why a smooth function type cannot automatically be used as a measurable function type for arbitrary stochastic programs.

16. From attention to sets and sequences. Vaswani and coauthors introduce the Transformer [53]. Zaheer and coauthors and Lee and coauthors develop architectures for sets [54, 55]. Bloem-Reddy and Teh relate invariant and equivariant networks to probabilistic symmetries [56]. The mask calculation here uses independent permutations of queries and key–value rows. It includes causal order by treating the mask as part of the data. An autoregressive policy additionally needs a rule for sampling and advancing the state.

17. Stochastic training and generation. Williams gives the REINFORCE gradient estimators [58]. Sutton and Barto give a systematic treatment of policy learning, prediction, and reinforcement learning [59]. Ho, Jain, and Abbeel give denoising diffusion probabilistic models [57]. Our discussion uses these as different sources of kernels or risk gradients. The differentiation and composition statements specify exactly which assumptions are needed to move an expectation past a derivative or to combine successive transitions.

18. Coalgebras and open systems. Rutten develops the general coalgebraic account of systems [61]. Vagner, Spivak, and Lerman study the operad of wiring diagrams for open dynamics [62]. Spivak relates polynomial functors to dynamical systems [63]. Open games supply one account of strategic composition [65]. Capucci and coauthors develop categorical cybernetics [66]. The stochastic agent in this paper

is specified by a kernel with an explicit state and a time step. The delayed feedback rule chooses an operational interpretation for cycles.

19. Observational equivalence. Finite-trace preservation is proved by induction on the joint state and history law. The external protocol can be adaptive as long as it uses only the specified external observations. This is related to the distinction between observed history and environmental state in partially observable decision processes [60]. Fritz and Rischel discuss infinite products in categorical probability [52]. The finite proof does not need an infinite path extension.

17.4 Distributed training and empirical scope

20. Federated local steps. McMahan and coauthors introduce FedAvg as a communication-efficient approach [67]. Karimireddy and coauthors analyze client drift and develop SCAFFOLD [68]. The two-step formula here is an exact quadratic calculation. It isolates one difference between synchronized central updates and averaged client trajectories. Descent of local models and statistical or algorithmic analysis of federated training are separate constructions.

21. Approximation and estimation. Cybenko and Hornik and coauthors give classical universal approximation results [69, 70]. Hoeffding's inequality gives the simple finite-class deviation bound quoted in Section 14 [71]. Both concern observable functions and specified analytic or statistical conditions. They do not identify the full model stack or the chosen optimizer. The distinction is useful when one wants to claim that two trainable programs are equivalent.

22. Other uses of sheaves and toposes in learning. Shiebler, Gavranović, and Wilson survey categorical approaches to learning [9]. Jia and coauthors provide a survey that explicitly includes topos-theoretic work [10]. The choice of site determines what locality means. For example neural sheaf diffusion uses vector spaces and restriction maps on a graph to define learned diffusion operations [11]. There the graph encodes the locality of data and features. Here the covers are covers of smooth test manifolds, and the local objects are weight assignments and their comparison maps. A graph architecture can be used as the predictor inside our moduli construction, but the graph sheaf and the model stack serve different purposes.

23. Comparison with the Maxwell construction. A preceding Maxwell formulation uses a homotopy fiber of an equation map on a stack of connections. The analogous operation here is the fiber of the risk differential over the zero covector. Both

account for local presentations and their isomorphisms. The variables and physical interpretations are different. A neural parameter symmetry need not be a gauge symmetry of a field theory, and a stationary equation need not determine an evolution law. The comparison is the categorical operation and the resulting distinction between local and global lifts.

17.5 How to read a claim of equivalence

The results give several concrete tests for a proposed identification. For a forward program, verify the parameter-change identity in (2.1). For a coordinate-independent update, verify the transported covectors, metric, and displacement rule. For a smooth family, check the overlap arrows and their cocycle. For a higher lift, check the obstruction class and specify the automorphisms retained by the presentation. For a derived critical calculation, specify the test category and the residuals whose Koszul complex is taken. For agents, identify the interface, environment protocol, initial law, and horizon before comparing history laws. These tests involve different data even when applied to one training pipeline.

The squared-neuron families exhibit a sharp distinction among the tests. Pointwise every target is exactly realized. Over a parameter circle the realizations glue in the effective quotient even though the corresponding bundle has no section. Over a projective plane the effective quotient still has an exact-fit family, but a specified ineffective extension has a nonzero class in degree two. These calculations connect predictive equivalence to the topology of training presentations and explain the role of the higher classifier.

References

[1] B. Fong, D. I. Spivak, and R. Tuyéras. Backprop as functor: A compositional perspective on supervised learning. *LICS*, 2019. Source

[2] B. Fong and M. Johnson. Lenses and learners. *Proceedings of the Eighth International Workshop on Bidirectional Transformations*, 2019. Source

[3] G. S. H. Cruttwell, B. Gavranović, N. Ghani, P. Wilson, and F. Zanasi. Categorical foundations of gradient-based learning. *ESOP*, 2022. Source

[4] R. Cockett, G. Cruttwell, J. Gallagher, J.-S. P. Lemay, B. MacAdam, G. Plotkin, and D. Pronk. Reverse derivative categories. *CSL*, 2020. Source

[5] B. Gavranović. *Fundamental Components of Deep Learning: A Category-Theoretic Approach*. PhD thesis, University of Strathclyde, 2024. Source

[6] B. Gavranović. Compositional deep learning. Preprint, 2019. Source

[7] G. S. H. Cruttwell, B. Gavranović, N. Ghani, P. Wilson, and F. Zanasi. Deep learning with parametric lenses. Preprint, 2024. Source

[8] 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. *ICML*, PMLR 235, 2024. Source

[9] D. Shiebler, B. Gavranović, and P. Wilson. Category theory in machine learning. Preprint, 2021. Source

[10] Y. Jia, G. Peng, Z. Yang, and T. Chen. Category-theoretical and topos-theoretical frameworks in machine learning: A survey. Preprint, 2024, revised 2025. Source

[11] C. Bodnar et al. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. *NeurIPS*, 2022. Source

[12] B. Fong and D. I. Spivak. *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*. Cambridge University Press, 2019. Source

[13] M. Riley. Categories of optics. Preprint, 2018. Source

[14] R. F. Blute, J. R. B. Cockett, and R. A. G. Seely. Cartesian differential categories. *Theory and Applications of Categories* 22 (2009), 622–672. Source

[15] T. Ehrhard and L. Regnier. The differential lambda-calculus. *Theoretical Computer Science* 309 (2003), 1–41. Source

[16] C. Elliott. The simple essence of automatic differentiation. *Proceedings of the ACM on Programming Languages* 2 (ICFP), 2018. Source

[17] B. A. Pearlmutter and J. M. Siskind. Reverse-mode AD in a functional framework: Lambda the ultimate backpropagator. *ACM Transactions on Programming Languages and Systems* 30 (2008). Source

[18] M. Abadi and G. D. Plotkin. A simple differentiable programming language. *Proceedings of the ACM on Programming Languages* 4 (POPL), 2020. Source

[19] 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 (2018), 1–43. Source

[20] M. Huot, S. Staton, and M. Vákár. Correctness of automatic differentiation via diffeologies and categorical gluing. *FoSSaCS*, 2020. Source

[21] W. Lee, S. Park, and A. Aiken. On the correctness of automatic differentiation for neural networks with machine-representable parameters. *ICML*, PMLR 202, 2023. Source

[22] S. Amari. Natural gradient works efficiently in learning. *Neural Computation* 10 (1998), 251–276. Source

[23] J. Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research* 21 (2020), 1–76. Source

[24] Y. Song, J. Song, and S. Ermon. Accelerating natural gradient with higher-order invariance. *ICML*, PMLR 80, 2018. Source

[25] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008. Source

[26] S. Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control* 58 (2013), 2217–2229. Source

[27] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp minima can generalize for deep nets. *ICML*, PMLR 70, 2017. Source

[28] C. Godfrey, D. Brown, T. Emerson, and H. Kvinge. On the symmetries of deep learning models and their internal representations. *NeurIPS*, 2022. Source

[29] M. Theiss, L. Braun, A. M. Saxe, and E. Grant. Parameter symmetries determine representational geometry in overparameterized nonlinear networks. Preprint, 2026. Source

[30] D. Kunin, J. Sagastuy-Breña, S. Ganguli, D. L. K. Yamins, and H. Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. *ICLR*, 2021. Source

[31] S. K. Ainsworth, J. Hayase, and S. Srinivasa. Git Re-Basin: Merging models modulo permutation symmetries. *ICLR*, 2023. Source

[32] R. Entezari, H. Sedghi, O. Saukh, and B. Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *ICLR*, 2022. Source

[33] J. Lurie. *Higher Topos Theory*. Annals of Mathematics Studies 170, Princeton University Press, 2009. Source

[34] S. Hollander. A homotopy theory for stacks. *Israel Journal of Mathematics* 163 (2008), 93–124. Source

[35] D. Dugger, S. Hollander, and D. C. Isaksen. Hypercovers and simplicial presheaves. *Mathematical Proceedings of the Cambridge Philosophical Society* 136 (2004), 9–51. Source

[36] T. Nikolaus, U. Schreiber, and D. Stevenson. Principal infinity-bundles: General theory. *Journal of Homotopy and Related Structures* 10 (2015), 749–801. Source

[37] T. Nikolaus, U. Schreiber, and D. Stevenson. Principal infinity-bundles: Presentations. *Journal of Homotopy and Related Structures* 10 (2015). Source

[38] J. Giraud. *Cohomologie non abélienne*. Grundlehren der mathematischen Wissenschaften 179, Springer, 1971. Source

[39] J.-L. Brylinski. *Loop Spaces, Characteristic Classes and Geometric Quantization*. Birkhäuser, 1993. Source

[40] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002. Source

[41] G. Vezzosi. Derived critical loci I: Basics. Preprint, 2011. Source

[42] T. Pantev, B. Toën, M. Vaquié, and G. Vezzosi. Shifted symplectic structures. *Publications Mathématiques de l'IHÉS* 117 (2013), 271–328. Source

[43] B. Toën. Derived algebraic geometry. *EMS Surveys in Mathematical Sciences* 1 (2014), 153–240. Source

[44] D. I. Spivak. Derived smooth manifolds. *Duke Mathematical Journal* 153 (2010), 55–128. Source

[45] F. W. Lawvere. The category of probabilistic mappings. Manuscript, 1962, published by the Lawvere Archive in 2025. Source

[46] M. Giry. A categorical approach to probability theory. In *Categorical Aspects of Topology and Analysis*, Lecture Notes in Mathematics 915, Springer, 1982. Source

[47] T. Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics* 370 (2020), 107239. Source

[48] B. Fong. Causal theories: A categorical perspective on Bayesian networks. Thesis, 2012, preprint 2013. Source

[49] K. Cho and B. Jacobs. Disintegration and Bayesian inversion via string diagrams. *Mathematical Structures in Computer Science* 29 (2019), 938–971. Source

[50] T. Fritz and P. Perrone. A probability monad as the colimit of spaces of finite samples. *Theory and Applications of Categories* 34 (2019), 170–220. Source

[51] C. Heunen, O. Kammar, S. Staton, and H. Yang. A convenient category for higher-order probability theory. *LICS*, 2017. Source

[52] T. Fritz and E. F. Rischel. Infinite products and zero-one laws in categorical probability. *Compositionality* 2 (2020), article 3. Source

[53] A. Vaswani et al. Attention is all you need. *NeurIPS*, 2017. Source

[54] M. Zaheer et al. Deep Sets. *NeurIPS*, 2017. Source

[55] J. Lee et al. Set Transformer: A framework for attention-based permutation-invariant neural networks. *ICML*, PMLR 97, 2019. Source

[56] B. Bloem-Reddy and Y. W. Teh. Probabilistic symmetries and invariant neural networks. *Journal of Machine Learning Research* 21 (2020), 1–61. Source

[57] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. Source

[58] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8 (1992), 229–256. Source

[59] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed., MIT Press, 2018. Source

[60] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101 (1998), 99–134. Source

[61] J. J. M. M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science* 249 (2000), 3–80. Source

[62] 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 (2015), 1793–1822. Source

[63] D. I. Spivak. Poly: An abundant categorical setting for mode-dependent dynamics. Preprint, 2020. Source

[64] D. I. Spivak. Compositional dynamics in learning and mechanics. Preprint, 2026. Source

[65] N. Ghani, J. Hedges, V. Winschel, and P. Zahn. Compositional game theory. *LICS*, 2018. Source

[66] M. Capucci, B. Gavranović, J. Hedges, and E. F. Rischel. Towards foundations of categorical cybernetics. *Applied Category Theory*, 2021. Source

[67] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. *AISTATS*, PMLR 54, 2017. Source

[68] S. P. Karimireddy et al. SCAFFOLD: Stochastic controlled averaging for federated learning. *ICML*, PMLR 119, 2020. Source

[69] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* 2 (1989), 303–314. Source

[70] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks* 2 (1989), 359–366. Source

[71] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* 58 (1963), 13–30. Source

[72] Macroquad contributors. Macroquad 0.4.16 API documentation, `Camera3D` and 3D drawing primitives. Source