Weight-Space Programming as Certified Network Editing: Higher Categorical Semantics and Architecture-Independent Generalization

Anonymous

November 2, 2026 Version 1.0

Abstract

We develop a mathematical semantics for direct inspection, synthesis, and editing of network parameters. A network is a parameterized interpretation with an explicit behavior map; an editor proposes changes, while a separate certificate family states what those changes establish. We distinguish behavior-preserving reparameterizations from directed behavior-changing edits using parameterized higher categories and certified spans. A compositional theorem transports certificates through typed wiring operations whenever the chosen behavioral relations are closed under those operations. Coherent symmetry descent removes coordinate artifacts, and effective-epimorphism descent gives an exact local-to-global criterion for certificates. The same construction applies to recurrent, graph, stochastic, Boolean, and hybrid networks, and to classical control of noncartesian backends. We prove identification obstructions, finite-query limits, quantitative perturbation bounds, and counterexamples to unrestricted edit composition. Exact-arithmetic examples accompany the manuscript. The results concern an explicit semantic model; they neither certify a language model's reasoning nor independently reproduce the motivating experiments.

Keywords: weight-space programming; parameterized categories; infinity-topos; operads; network editing; descent; certificates; agentic meta-algorithms.

1 Question, source, and contribution

Takehi, Teoh, Zhang, and Khattab introduce weight-space programming as direct reading and writing of neural parameters by a general-purpose language model, sometimes with restricted execution feedback [1]. They report examples of inspection, repair, and synthesis, together with unsuccessful cases. We treat their article as the motivating empirical report, not as a theorem or an independently reproduced benchmark. Our question is mathematical: what must be specified before an edit can carry a compositional claim about behavior, and which parts of that specification survive a change of network architecture?

The contribution is a certificate-based organization of that question. It separates (i) the coordinates stored in a checkpoint, (ii) the program denoted by those coordinates, (iii) the observations available to an editor, and (iv) evidence for the requested change. Existing parameterized categories, wiring semantics, and higher-topos descent supply the underlying mathematics [2, 3, 6, 8]. We do not claim these constructions as new. The paper combines them into a precise protocol for direct network editing and identifies the assumptions required for its architecture-independent extension.

The principal result, Theorem 6.1, states an explicit closure condition under which certified component edits assemble coherently. This condition matters: two individually beneficial edits can

be harmful together. The later results explain what quotient symmetries, local certificates, finite evaluation, and implementation error do and do not justify. All proofs are ordinary mathematical proofs. The accompanying Python program checks finite examples; it is not a proof-assistant formalization.

2 Semantic universe and network presentations

Fix universe bounds and an infinity-topos $\mathcal{E}$. Every object, product, and diagram below is small in the relevant bound. Write Y^X for an internal exponential and $\mathrm{Map}_{\mathcal{E}}(X, Y)$ for an external mapping space. Limits and pullbacks are homotopy-coherent. The internal language is dependent and intuitionistic unless a particular example explicitly uses classical sets.

Definition 2.1 (Presented network)**.** A network presentation N with interface (X_N, Y_N) consists of a parameter object Θ_N and an evaluation morphism

$$e_N : \Theta_N \times X_N \longrightarrow Y_N.$$

Its transpose $\beta_N : \Theta_N \to Y_N^{X_N}$ is its behavior map. More generally, one may specify a behavior object $\mathcal{B}_N$ and a morphism $\beta_N : \Theta_N \to \mathcal{B}_N$; examples include transition systems, stochastic kernels, or observable traces rather than single-step functions.

Architecture, preprocessing, masks, input meanings, numeric format, execution order, and decoding belong to the presentation. A tensor of numbers without these data does not determine an input-output program. Finite-precision execution is a separate evaluation morphism, not automatically the same morphism as ideal real arithmetic.

A necessary distinction about spaces. In $\mathcal{E} = \mathcal{S}$, use real or rational numbers as *0-truncated objects* when exact numeric values are intended. Replacing the real line by its ordinary homotopy type makes it contractible and loses the distinctions needed here. Topological continuity does not make two real-valued functions equal, and a continuous interpolation between parameter vectors is not a behavior-preservation certificate. Smoothness, order, and metrics are additional structure, not consequences of merely choosing an infinity-topos. Our exact examples use discrete rational arithmetic. Higher structure will arise from symmetry groupoids and coherent certificate data.

Definition 2.2 (Specification and evidence)**.** A property of behaviors is a subobject $S \hookrightarrow \mathcal{B}_N$. A structured certificate family is any map $q : C \to \mathcal{B}_N$, together with a specified factorization through S when soundness for that property is claimed. The object of certified parameters is

$$V_N = \Theta_N \times_{\mathcal{B}_N} C. \tag{1}$$

A section over a particular parameter is evidence for its membership. If q is a monomorphism, its fibers are propositions. A general q may retain multiple witnesses and nontrivial higher identifications.

Natural-language instructions must be translated into such a specification before they acquire a truth condition in this model. An editor's explanation alone does not supply the factorization asserting soundness. For deployment, one must additionally bind the certificate to the exact presentation and artifact, including preprocessing and execution semantics.

3 Parameterized higher categories and directed edits

3.1 Behavior-preserving reparameterization

For fixed X, Y, consider the presheaf on $\mathcal{E}$

$$P \longmapsto \mathrm{Map}_{\mathcal{E}}(P \times X, Y).$$

Its Grothendieck construction is an infinity-category $\mathrm{Param}_{\mathcal{E}}(X, Y)$. An object is $(P, f : P \times X \to Y)$. A morphism from (P, f) to (Q, g) consists of $a : P \to Q$ and a specified homotopy

$$f \simeq g \circ (a \times \mathrm{id}_X). \tag{2}$$

Higher simplices retain the compatibilities between these homotopies. In ordinary sets this is the familiar commuting-triangle definition of a reparameterization [3].

Horizontal composition is

$$(Q, g) \circ (P, f) = (P \times Q,\ (p, q, x) \mapsto g(q, f(p, x))). \tag{3}$$

The identity is $(1, \mathrm{id}_X)$. Products and function composition give coherent associators, units, and interchange, organizing these hom infinity-categories into a parameterized $(\infty, 2)$-category. For example, reparameterizations a and b compose horizontally as $a \times b$; applying their two specified homotopies proves the required commuting triangle. Associating three factors in either order yields the same evaluation after the canonical product associator. Its coherence is inherited from cartesian products in $\mathcal{E}$. This is the higher version of standard parameterized-map semantics [2, 3].

Remark 3.1 (An edit is usually not a reparameterization)**.** Equation (2) preserves the represented behavior. A repair generally changes it. Therefore a weight edit must not be declared a higher equality merely because its endpoints have the same architecture or nearby parameter values.

3.2 Certified spans for behavior change

Fix $\beta : \Theta \to \mathcal{B}$ and a behavioral relation, or relation with witnesses,

$$r : R \longrightarrow \mathcal{B} \times \mathcal{B}.$$

Define the universal certified-edit object

$$\widehat{R} = (\Theta \times \Theta) \times_{\mathcal{B} \times \mathcal{B}} R. \tag{4}$$

A certified edit family is a map $K \to \widehat{R}$, equivalently a span

$$\Theta \xleftarrow{s} K \xrightarrow{t} \Theta$$

with a specified lift of $(\beta s, \beta t)$ to R. This describes a directed change and its evidence. It does not assert equality of the endpoints.

If relations R_{01} and R_{12} have a specified relational-composition witness

$$R_{01} \times_{\mathcal{B}} R_{12} \longrightarrow R_{02}, \tag{5}$$

then edit families K_{01} and K_{12} compose using $K_{01} \times_{\Theta} K_{12}$. The matching point in the homotopy pullback supplies the required identification of the intermediate behavior. Equation (5) then supplies the composite certificate. Associators come from equivalences of iterated pullbacks. When relation

composition has coherent units and associativity, certified spans inherit those coherences. Maps of spans preserving their lifts are 2-morphisms; higher identifications between those maps are retained. Thus the model includes directed revisions, alternative witnesses, and coherent transformations between revision protocols.

Examples of R include equality of observable behavior, preservation on a protected domain, an error bound $d(b, b') \leq \epsilon$, or a transition satisfying a safety contract. A relation saying only "the utility increased from this baseline" has no automatic closure under arbitrary network wiring.

4 Reading, synthesis, editing, and permitted effects

4.1 Reading is an inference problem

Let $o : \Theta \to O$ describe the information actually available: raw weights, selected tensors, architecture metadata, or an allowed transcript. Let $\ell : \Theta \to L$ be the desired answer, such as a behavioral classification. A correct exact reader is a factorization $d \circ o = \ell$. A statistical reader instead produces a distribution of answers; confidence scores require a separate calibration criterion.

Proposition 4.1 (Identification criterion in sets)**.** *Let $o : \Theta \twoheadrightarrow O$ be surjective and L a set. A function $d : O \to L$ with $do = \ell$ exists if and only if*

$$o(\theta) = o(\eta) \quad \Longrightarrow \quad \ell(\theta) = \ell(\eta).$$

Proof. Necessity follows by applying d. Conversely the displayed condition gives each fiber of o a unique label; define d by that label. The definition is independent of the chosen representative. If o is not surjective, the same conclusion holds on its image; extending beyond the image requires additional choices or an inhabited answer set. □

For an effective epimorphism in an infinity-topos, the corresponding statement is descent: the proposed reader must carry coherent matching data along the entire Cech nerve of o. Fiberwise equality suffices for the set-valued case; higher-valued answers can require higher compatibility. Missing input encodings can therefore prevent identification even when the numerical weights are fully available.

Corollary 4.2 (Statistical indistinguishability)**.** *If two possible systems have the same law of allowed observations, every reader depending only on those observations has the same answer law for both. For two distinct binary labels with equal prior probabilities, its average classification success is at most* $1/2$*.*

Proof. Push the common observation law through the reader's stochastic kernel. If its probability of answering the first label is p, its average success is $\frac{1}{2}p + \frac{1}{2}(1 - p) = \frac{1}{2}$; abstention cannot increase this value. □

4.2 Synthesis and editing are proposal mechanisms

A synthesizer takes a formal specification and proposes a point of Θ, possibly returning failure. A certified synthesizer returns a point of V_N in (1). An editor additionally takes an existing parameter and proposes an endpoint of (4). Neither definition implies that an arbitrary requested behavior is representable by the chosen architecture.

In a concrete finite stochastic model, write an editor step as a kernel

$$H \times \mathrm{Spec} \times \Theta \longrightarrow \mathsf{D}\big(H \times (\Theta + \{\mathrm{fail}\}) \times \mathbb{N}^d\big),$$

where H is durable editor state, D is the finite-distribution monad, and the resource vector records permitted calls, batch sizes, generated tokens, or elapsed steps. Tools have typed signatures and return transcripts. Sequential execution adds the resource vectors. This is an operational model in sets, not an assertion that every infinity-topos carries this particular probability monad.

An effect discipline can prohibit gradient queries or training loops while permitting tensor algebra and a bounded number of executions. Such restrictions concern the *implementation trace*: extensionally equal updates can be produced by different procedures. The phrase "no gradient descent" by itself neither proves correctness nor excludes every form of search. A complete report must specify what constitutes a call and what computations each call includes.

Proposition 4.3 (Acceptance soundness)**.** *If a checker accepts a proposed (θ, θ') only by producing a point of (4), and the certificate family factors through the intended property relation, every accepted edit satisfies that relation in the chosen semantics.*

Proof. Compose the returned lift with its specified soundness factorization. The pullback square defining $\widehat{R}$ identifies its endpoints with $(\beta\theta, \beta\theta')$. □

This proposition isolates the obligation; it does not construct a complete checker. A checker that establishes only finitely many test outcomes certifies that finite specification.

5 Coordinate symmetries and higher quotients

Let a group object G act on Θ, with a coherently invariant behavior map $\beta : \Theta \to \mathcal{B}$ and trivial action on $\mathcal{B}$. The action-groupoid colimit $\Theta//G$ retains stabilizers and their identifications.

Theorem 5.1 (Symmetry descent; standard quotient property)**.** *The behavior map factors through $\bar{\beta} : \Theta//G \to \mathcal{B}$. Every coherently invariant reader $\Theta \to L$ similarly descends. A coherently G-equivariant edit $a : \Theta \to \Theta$ induces an edit $\bar{a}$ of the quotient.*

Proof. Mapping out of the colimit of the action groupoid is the limit of the mapping spaces out of its simplicial levels. A coherent invariance datum is exactly a point of this limit, yielding the first two factorizations. An equivariant edit induces a map of action groupoids; functoriality of colimits gives $\bar{a}$. The resulting squares commute with their specified coherences. □

For a feedforward network with coordinatewise activations, let S_i be permutation matrices on hidden layers and $S_0 = S_L = I$ at the external ports. Transform

$$W_i' = S_i W_i S_{i-1}^{-1}, \qquad b_i' = S_i b_i. \tag{6}$$

Since $\sigma_i(S_i z) = S_i \sigma_i(z)$, induction gives $h_i' = S_i h_i$ and identical final outputs. This is the permutation symmetry exploited in deep weight-space learning [4]. Arbitrary invertible changes of basis do not generally commute with nonlinear activations.

The quotient need not be a set. If a two-unit presentation is fixed by swapping its units, its quotient point retains the automorphism group S_2; in the extreme fixed-point example it is BS_2. The orbit set forgets that automorphism. Consequently an explanation assigning a unique identity to one of two identical hidden units requires a choice of presentation. Behavioral explanations may instead descend to the quotient.

Equation (6) is exact for the stated mathematical arithmetic. Floating-point summation can depend on term order, so a physical implementation requires either an order-preserving execution argument or an explicit error bound. Sparse-edit counts and Euclidean distances are also presentation-dependent; neither is itself a semantic certificate.

6 Typed wiring and architecture-independent certificates

Let $\mathcal{O}$ be a small colored operad, ordinary or higher, specifying legal network assembly. A color is an external interface; an operation $w : (c_1, \dots, c_k) \to c$ is a typed wiring context. Only operations whose semantics have been supplied are included. In particular, arbitrary feedback is not licensed merely by drawing a cycle.

Assume parameter objects Θ_c and behavior objects $\mathcal{B}_c$ form $\mathcal{O}$-algebras in $\mathcal{E}$, with assembly maps a_w and b_w. A behavior interpretation is a morphism of those algebras, meaning coherently

$$\beta_c \, a_w \simeq b_w \prod_{i=1}^{k} \beta_{c_i}. \tag{7}$$

For instance, a color (X, Y) can have $\Theta_c = \coprod_{N:c} \Theta_N$ over a small set of architectures, and $\mathcal{B}_c = Y^X$. Sequential and parallel composition induce operations. Fixed-architecture parameter objects are particular summands. Parameter sharing is represented by a common control object W and maps $W \to \Theta_{c_i}$, not by an assumption of independently editable coordinates.

Choose relation families $R_c \to \mathcal{B}_c \times \mathcal{B}_c$. A certificate assembly for w is a map

$$\rho_w : \prod_i R_{c_i} \longrightarrow R_c \tag{8}$$

whose endpoint maps are the two applications of b_w. Require these maps to satisfy the operadic units, symmetries, and substitution laws with coherent witnesses. This is the precise closure hypothesis.

Theorem 6.1 (Compositional certified editing)**.** *Under* (7) *and* (8)*, every wiring w induces*

$$\widehat{a}_w : \prod_i \widehat{R}_{c_i} \longrightarrow \widehat{R}_c.$$

Its endpoints are the assemblies of the original and edited components. These induced maps obey the operadic coherences. The construction also holds after restriction along any shared-parameter control object.

Proof. Each factor $\widehat{R}_{c_i}$ maps to both copies of Θ_{c_i} and to R_{c_i}. Apply a_w to the two tuples of parameter endpoints and ρ_w to the tuple of witnesses. Equation (7) and the endpoint condition in (8) give the homotopy-commuting square

$$\begin{array}{ccc} \prod_i \widehat{R}_{c_i} & \longrightarrow & R_c \\ \downarrow & & \downarrow \\ \Theta_c \times \Theta_c & \xrightarrow[\beta_c \times \beta_c]{} & \mathcal{B}_c \times \mathcal{B}_c. \end{array}$$

The pullback universal property gives $\widehat{a}_w$. Substituting wirings before or after this construction produces cones whose coherences are those already specified by the two algebras and the relation families. The space of compatible lifts of each such cone is contractible. Hence the induced maps inherit all these coherences. Pulling this diagram back along a shared-control map proves the last assertion. □

This theorem concerns a single witnessed tuple of edits, not arbitrary independent updates to tied parameters. The common-control condition must hold for the proposed endpoints as well as the original ones. The theorem applies without differentiability, a loss function, or a particular proposal algorithm. It gives a sufficient condition for certificate transport, not a universal algorithm for finding the certificates.

Example 6.2 (Equality and contextual preservation)**.** Behavior equality is closed under every defined wiring operation. Thus exact component equivalences compose. Preservation restricted to an input domain composes only if every intervening component maps that domain into the domain protected by the next certificate. Omitting this range condition invalidates the conclusion.

Example 6.3 (Individually beneficial edits need not compose)**.** For $F(a, b) = a + b - 3ab$, use larger output as utility. From $(0, 0)$, changing either coordinate alone to 1 improves the output from 0 to 1. Changing both gives $F(1, 1) = -1$. Therefore baseline utility improvement does not satisfy (8) for this wiring. An editor needs an interaction certificate or a joint check.

6.1 What makes the generalization higher categorical?

Parameterized maps retain hom infinity-categories; directed edits compose by homotopy pullbacks; quotient presentations retain stabilizer groupoids; certificate transport retains the coherences of an operad algebra. These are different roles. None licenses treating a directed improvement as an equivalence. If all objects and witnesses are discrete and symmetries are discarded, the construction reduces to an ordinary compositional semantics. That useful special case does not by itself demonstrate nontrivial higher structure.

7 Extension to other network families

The theorem depends on typed assembly and sound relation closure, not on the word "neural." Table 1 lists concrete instantiations. Each row requires its own behavior interpretation and checker; no universal network semantics is assumed.

Network family	Parameter and behavior interpretation	Required additional structure
Feedforward / convolutional	Kernels and biases; finite input-output maps	Convolutional sharing, boundary and activation conventions
Transformer	Attention and MLP tensors; bounded token-to-distribution map	Tokenizer, masks, positions, cache state, and decoding fixed
Graph network	Shared update rules; graph-indexed equivariant maps	Permutation-compatible aggregation and readout
Recurrent / control	Transition and output maps; finite or infinite traces	Initial state and a justified causal feedback operator
Stochastic / diffusion	Parameterized transition kernels; path or output laws	Measurability, noise law, and kernel composition
Boolean / symbolic	Gates, tables, or programs; discrete functions or traces	Exact gate semantics; termination restrictions when needed
Hybrid / logistics	Discrete modes and continuous or timed transitions	Event ordering, well-posed flow, and environment assumptions
Tensor / quantum backend	Classical controls choosing tensor or channel morphisms	Backend composition and tensor; no unrestricted quantum copying

Table 1: Conditional architecture-independent instantiations.

Graph networks. For a finite graph A, a typical layer has

$$h'_v = \phi_\theta \left(h_v, \sum_{u \in N_A(v)} \psi_\theta(h_u, h_v) \right).$$

A graph isomorphism permutes the summands and vertices. In exact arithmetic, invariant aggregation and shared ϕ_θ, ψ_θ give an equivariant layer. Inducting over layers proves an equivariant network; an invariant readout gives an invariant graph-level output. An edit to the shared rules preserves this symmetry when their types and sharing remain fixed. Numerical accuracy or task success is a different relation requiring separate evidence.

Recurrent and hybrid networks. A state transition $T_\theta : S \times X \to S$ and output map $O_\theta : S \times X \to Y$ determine a finite-horizon behavior by recursion from a specified initial state. Infinite traces require the corresponding limit or a guarded/causal construction. For continuous-time components, existence and uniqueness of a flow must be supplied before using it as an operation. Algebraic loops may have no solution or multiple solutions, so arbitrary instantaneous feedback is excluded. The wiring viewpoint follows established open-system semantics [6].

Stochastic networks. In the finite case, $\mathcal{B}_{X,Y}$ can be the set of Markov kernels $X \to \mathsf{D}(Y)$. Sequential composition is

$$(L \circ K)(z \mid x) = \sum_y L(z \mid y) K(y \mid x).$$

Independent parallel composition uses the product kernel; correlated randomness must be represented explicitly as a shared input or joint kernel. Kernel-valued behavior objects can be treated as discrete objects of $\mathcal{S}$ while their composition remains noncartesian stochastic semantics. This is consistent with the Markov-category perspective [7]; it does not assert that the category of stochastic maps is itself an infinity-topos.

Noncartesian backends. For a small symmetric monoidal backend $\mathcal{V}$, choose behavior objects presenting its hom spaces and define wiring by composition and tensor. Classical parameter objects select those morphisms. Theorem 6.1 then concerns the externally parameterized behavior algebra. It does not turn $\mathcal{V}$ into a cartesian category or introduce an unavailable diagonal $Q \to Q \otimes Q$ for an arbitrary quantum object. A tensor-network rewrite that is an exact morphism equality and a physical parameter edit that changes a channel are different certificates.

Proposition 7.1 (Transfer between semantic universes)**.** *Let $F : \mathcal{E} \to \mathcal{F}$ preserve finite limits. Applying F to the parameter, behavior, and certificate-assembly diagrams transports every certified-edit construction above to the corresponding $F\mathcal{B}$-valued interpretation in $\mathcal{F}$.*

Proof. Finite-limit preservation identifies $F\widehat{R}$ with the pullback of FR along $F\Theta \times F\Theta \to F\mathcal{B} \times F\mathcal{B}$. It also preserves the finite products in each assembly diagram and the given homotopies and coherences. Applying F to the proof of Theorem 6.1 therefore gives the required lift. □

This is a diagram-transfer statement. A geometric inverse image need not preserve exponentials: one cannot generally replace $F(Y^X)$ by $(FY)^{FX}$. Identifying them requires an additional hypothesis. Numeric inequalities, probability laws, and infinite trace constructions likewise require preservation of their particular structures beyond the stated finite-limit result.

8 Quantitative certificates

Metric arguments are external classical specializations with the following stated bounds. They are not automatic consequences of the internal logic.

Theorem 8.1 (Sequential perturbation bound)**.** *Let $f_i, g_i : D_{i-1} \to D_i$ be maps between metric spaces. Suppose g_i is L_i-Lipschitz and $d(f_i(x), g_i(x)) \le \delta_i$ for every $x \in D_{i-1}$. Then*

$$\sup_{x \in D_0} d(f_n \cdots f_1(x), g_n \cdots g_1(x)) \le \sum_{i=1}^{n} \delta_i \prod_{j=i+1}^{n} L_j. \tag{9}$$

Proof. Use the hybrid compositions $H_i = g_n \cdots g_{i+1} f_i \cdots f_1$ for $0 \le i \le n$, interpreting empty strings as identities. Consecutive hybrids differ only at stage i. Their distance is at most δ_i multiplied by the Lipschitz constants of the common downstream g maps. Summing the n distances proves the claim. □

The domains must contain every intermediate value used in this proof. With a protected subset, closure and range certificates supply this requirement. For example, a first layer perturbation $1/4$ followed by a 3-Lipschitz layer and an additional second-layer error $1/5$ yields $19/20$, not $1/4 + 1/5$.

Corollary 8.2 (Contractive recurrent editing)**.** *Suppose $T, T' : S \times X \to S$ share an initial state, T' is uniformly L-Lipschitz in state, and $d(T(s,x), T'(s,x)) \le \delta$ on an invariant domain. Under the same input sequence,*

$$d(s_t, s'_t) \le \delta \sum_{k=0}^{t-1} L^k.$$

If $L < 1$, the bound is uniformly at most $\delta/(1-L)$. A common K-Lipschitz state readout multiplies the bound by K.

Proof. The triangle inequality gives $e_{t+1} \le \delta + L e_t$, with $e_0 = 0$. Induction solves this recurrence. Sum the geometric series when $L < 1$. □

For $s_{t+1} = s_t/2 + x_t$ and an edit adding $1/8$ to each transition, the exact difference at $t = 4$ is $15/64$ and its limiting bound is $1/4$. No analogous uniform contractive conclusion follows when $L \ge 1$.

Proposition 8.3 (Finite-kernel error propagation)**.** *If $K, K' : X \to \mathsf{D}(Y)$ and $L, L' : Y \to \mathsf{D}(Z)$ satisfy uniform total-variation bounds ϵ and η, respectively, then*

$$\sup_x \mathrm{TV}(LK(x), L'K'(x)) \le \epsilon + \eta.$$

Proof. Insert $LK'(x)$ and use the triangle inequality. Applying a stochastic kernel contracts total variation: expanding the finite sums bounds the ℓ^1 difference by the original ℓ^1 difference. For the second term, convexity bounds the distance between the $K'(x)$-mixtures of $L(y)$ and $L'(y)$ by η. Add the two bounds. □

Here $\mathrm{TV}(p, q) = \frac{1}{2} \sum_y |p_y - q_y|$. The bound can be capped by 1. This supplies a relation-closure witness for finite sequential stochastic networks. It does not establish small total variation between arbitrary language-model output laws from a small checkpoint edit.

9 Local-to-global certificates in an infinity-topos

Fix $q : C \to \mathcal{B}$, $\beta : \Theta \to \mathcal{B}$, and a family of candidate parameters $\theta : W \to \Theta$. Pulling back (1) gives a certificate object $V_W \to W$. Let $p : U \to W$ be an effective epimorphism and $U_\bullet$ its Cech nerve.

Theorem 9.1 (Certificate descent; standard topos consequence)**.** *Sections of $V_W \to W$ are equivalent to coherent matching sections of its pullbacks over all levels of $U_\bullet$. A fixed complete matching datum determines a global section up to a contractible space of choices.*

Proof. In the slice infinity-topos $\mathcal{E}_{/W}$, effectiveness says that the colimit of $U_\bullet$ is its terminal object W. Mapping this colimit into V_W turns it into the limit

$$\mathrm{Map}_{\mathcal{E}_{/W}}(W, V_W) \simeq \mathrm{Tot}\,\mathrm{Map}_{\mathcal{E}_{/W}}(U_\bullet, V_W).$$

The right side is precisely the space of matching sections and all their coherences. The homotopy fiber of this equivalence over a fixed datum is contractible. □

The effective-descent foundation is standard higher topos theory [8]. Its use here is to make the missing obligation explicit: locally produced structured certificates must agree on overlaps, their agreements must agree on triple overlaps, and so on. For a proposition-valued certificate, higher witness choices are contractible and property membership is local. For general certificate objects, nontrivial matching data can remain.

A finite benchmark is not generally an effective-epimorphism cover of an entire input domain. Nor do independent statements about distinct layers automatically provide a common certificate family with the required overlap data. The theorem therefore does not turn incomplete testing or unrelated local explanations into a global guarantee. Symmetry quotienting can remove a requirement to choose a global hidden-unit ordering, but it does not manufacture a task-success certificate.

10 Limits on universal identification and validation

Proposition 10.1 (Finite oracle queries do not certify an unseen point)**.** *Let X be finite and consider every Boolean function $X \to \{0, 1\}$ as an allowed behavior. A deterministic validator with only value-oracle access and fewer than $|X|$ queries on every run cannot both accept the zero function and soundly certify that a function is zero everywhere.*

Proof. Run it against the zero function. An unqueried x_* remains. The function that is 1 only at x_* gives the same answers to the adaptive queries, so the entire transcript and acceptance decision are unchanged. That acceptance is unsound. □

This is an oracle-model limit. It does not apply to a symbolic verifier that also inspects a program or the entire weight representation and proves facts about it. A fixed finite input domain can be checked exhaustively. It is incorrect to use this proposition to forbid all useful weight-based reasoning.

Proposition 10.2 (No complete terminating reader for unrestricted programs)**.** *If a generalized network family effectively represents arbitrary programs with unbounded execution, no total computable reader decides for every presentation whether its trace eventually emits a distinguished symbol.*

Proof. Given a machine and input, construct a network that simulates it and emits the symbol exactly when it halts. A proposed total reader would decide the halting problem, which is impossible [11]. □

The unbounded-program hypothesis is essential. We make no such undecidability claim for every fixed-size feedforward model over a finite domain.

Proposition 10.3 (Fresh-validation bound)**.** *An editor proposes at most m candidates. For each candidate, conditional on the entire previous history and that candidate, a validator draws n fresh independent Bernoulli trials from a fixed task distribution. Let p_j and $\widehat{p}_j$ be its true and empirical success probabilities. With probability at least $1-\alpha$, every evaluated candidate satisfies $|\widehat{p}_j - p_j| \le \epsilon$ provided*

$$n \ge \frac{\log(2m/\alpha)}{2\epsilon^2}.$$

Proof. Conditional Hoeffding inequality [10] bounds the failure probability at step j by $2e^{-2n\epsilon^2}$. Integrating over the prior history keeps that bound. A union bound over at most m steps gives $2me^{-2n\epsilon^2} \le \alpha$. A candidate selected among these steps inherits the simultaneous bound. □

Reusing a holdout while adapting candidates to its outcomes does not meet the fresh conditional-independence hypothesis. A distributional confidence statement also does not certify a universal safety property. These distinctions apply equally to neural and non-neural editing agents.

11 Worked examples and reproducibility

The package contains `checks/exact_checks.py`, runnable with standard Python 3. It uses `fractions.Fraction` for rational network values and finite-kernel probabilities. A JSON report records named checks and exact witnesses.

The checks cover a two-layer ReLU network under hidden-unit permutation; a sparse repair $f(x) = \mathrm{ReLU}(x)$ to $g(x) = \mathrm{ReLU}(x) + \mathrm{ReLU}(-x) = |x|$; relational preservation on $x \ge 0$; sequential error propagation; shared-rule graph equivariance; a contractive recurrence; finite stochastic-kernel propagation; a finite-cover gluing example and an incompatible overlap; an observation-label obstruction; the utility-interference example; and the unseen-input oracle counterexample.

For the absolute-value repair, the identity for all rational x follows by splitting into $x \ge 0$ and $x \le 0$. For nonnegative inputs, the newly active branch vanishes, proving protected-domain preservation. The finite executable enumeration illustrates these identities and checks the particular implementation, but does not replace those universal arguments. Likewise the finite-cover example demonstrates set-valued matching; it does not implement arbitrary higher Cech descent.

Claim boundary. The package contains no trained model, large-language-model invocation, GPU benchmark, verified compiler, or proof-assistant kernel certificate. It introduces no new empirical comparison with the source experiments. Re-running the finite checks validates the recorded witnesses for the supplied implementation. Checking the general theorems still requires reading their hypotheses and mathematical proofs.

12 Related work and practical interpretation

Parameterized learning already has compositional formulations: Fong, Spivak, and Tuyeras model supervised learning and backpropagation categorically, while Cruttwell and collaborators develop semantics with parameterized maps, lenses, and reverse derivatives [2, 3]. Our editor is deliberately independent of a reverse derivative: it may propose a direct update, but acceptance still requires a behavior relation. This does not imply that proposal mechanisms are equally effective.

Deep weight-space architectures exploit hidden-neuron permutation symmetries [4]. We state the descent obligation for readers and editors and preserve stabilizer information. Tracr compiles specified human-readable programs into transformer weights [5]; that is a useful contrast with

searching for edits to a preexisting checkpoint. Direct parameter synthesis is correct only relative to the same kind of specified interpretation, whether a compiler, a human, or a language model proposes the parameters.

Wiring algebras provide the established syntax for open-system assembly [6], while Markov categories distinguish deterministic copying from stochastic composition [7]. Higher topos theory supplies coherent quotients and descent [8], and higher operad theory organizes coherent substitution [9]. Their role is structural: none entails that natural-language explanations are faithful, that all requests are realizable, or that empirical success transfers across domains.

An implementable editing service should store the exact network presentation; the requested behavior relation; permitted tools and their resource accounting; original and proposed artifact identifiers; the checker and its assumptions; and the returned witnesses. Hashes bind files but do not prove their semantics. A correct checker can be small while the proposing agent remains heuristic. Rejected proposals and explicitly unknown obligations are legitimate outcomes.

The resulting generalization is precise: whenever a network family admits a typed behavior algebra and the chosen certificates are closed under its wiring, the same assembly theorem applies. For recurrent, probabilistic, or hybrid systems, supplying that algebra includes the well-posedness, causality, and distributional assumptions stated above. This is the extent of the claimed architecture independence.

Authorship, automation, and evidence

Prepared as an anonymous manuscript at the requesting user's direction with substantial AI assistance in exposition, mathematical development, code generation, and artifact preparation. The human requester is the responsible submitting party; no independent referee report is asserted. The motivating empirical claims belong to Takehi and collaborators. All other references below identify primary mathematical or research sources. Links and metadata were checked during preparation on November 2, 2026 (America/Los_Angeles). Mathematical claims are relative to the explicit hypotheses; successful finite tests are not represented as machine-checked universal proofs. This document is prepared for editorial consideration by the Meta Journal of Agentic Meta Algorithms; it does not assert acceptance or publication.

References

[1] R. Takehi, J. Teoh, A. Zhang, and O. Khattab. *Sparks of Weight-Space Programming: LLMs can natively read & write neural network weights*. Research article, November 2, 2026. https://rikiyat.github.io/blog/weight-space-programming/.

[2] B. Fong, D. I. Spivak, and R. Tuyeras. *Backprop as Functor: A compositional perspective on supervised learning*. LICS, 2019; preprint arXiv:1711.10455. https://arxiv.org/abs/1711.10455.

[3] G. S. H. Cruttwell, B. Gavranovic, N. Ghani, P. Wilson, and F. Zanasi. *Categorical Foundations of Gradient-Based Learning*. arXiv:2103.01931, 2021. https://arxiv.org/abs/2103.01931.

[4] A. Navon, A. Shamsian, I. Achituve, E. Fetaya, G. Chechik, and H. Maron. *Equivariant Architectures for Learning in Deep Weight Spaces*. Proceedings of Machine Learning Research 202, 25790–25816, 2023. https://proceedings.mlr.press/v202/navon23a.html.

[5] D. Lindner, J. Kramar, S. Farquhar, M. Rahtz, T. McGrath, and V. Mikulik. *Tracr: Compiled Transformers as a Laboratory for Interpretability*. arXiv:2301.05062, 2023. https://arxiv.org/abs/2301.05062.

[6] 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; arXiv:1408.1598. https://arxiv.org/abs/1408.1598.

[7] T. Fritz. *A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics*. Advances in Mathematics 370, 107239, 2020; arXiv:1908.07021. https://arxiv.org/abs/1908.07021.

[8] J. Lurie. *Higher Topos Theory*. Annals of Mathematics Studies 170, Princeton University Press, 2009; author's updated version. https://www.math.ias.edu/~lurie/papers/HTT.pdf.

[9] J. Lurie. *Higher Algebra*. September 18, 2017. https://www.math.ias.edu/~lurie/papers/HA.pdf.

[10] W. Hoeffding. *Probability Inequalities for Sums of Bounded Random Variables*. Journal of the American Statistical Association 58(301), 13–30, 1963. https://doi.org/10.1080/01621459.1963.10500830.

[11] A. M. Turing. *On Computable Numbers, with an Application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society s2-42(1), 230–265, 1937 (received 1936). https://doi.org/10.1112/plms/s2-42.1.230.