A Color-Geometry Causal Framework for Gradient-Free Learning in Spiking Neural Networks

I. Introduction: From Spikes to Spectra, From Gradients to Geometry

1.1. The Neuromorphic Impasse and the Fragility of Gradients

The field of neuromorphic computing stands at a critical juncture, promising a paradigm of energy-efficient, event-driven computation inspired by the brain's architecture. Spiking Neural Networks (SNNs), the cornerstone of this paradigm, leverage sparse, asynchronous communication to process information, making them theoretically ideal for low-power, low-latency applications. However, this promise has been persistently hindered by a fundamental challenge: the effective training of these networks. The immense success of conventional deep learning is built upon the foundation of backpropagation, an algorithm that excels at computing gradients in systems of continuous, differentiable functions. SNNs, in stark contrast, operate on discrete, all-or-none spike events, which are inherently non-differentiable and thus create a profound incompatibility with the mathematical machinery of gradient-based optimization.

This incompatibility manifests in a series of well-documented challenges. The spike generation mechanism itself, mathematically modeled as a Heaviside step function, has a derivative that is zero almost everywhere and undefined at the firing threshold. This gives rise to the "dead neuron" problem, where neurons that fail to reach their threshold provide no gradient signal for learning, effectively becoming inert. To circumvent this, the most prevalent training methods rely on surrogate gradients, which substitute the true derivative with a smooth proxy during the backward pass. While enabling significant progress, this approach introduces a "gradient mismatch" between the network's actual forward dynamics and the error signal used for learning, a compromise that can limit performance and requires the high memory and computational overhead of Backpropagation Through Time (BPTT). More recent developments in exact gradient methods offer mathematical precision by analytically calculating the effect of parameter changes on spike timing. Yet, this precision comes at the cost of high computational complexity and often requires access to dense state information, such as the continuous membrane potential of every neuron at all times—a requirement fundamentally at odds with the sparse, event-driven principles of neuromorphic hardware. These persistent issues suggest a deep paradigm mismatch: current methods are sophisticated attempts to force an inherently discrete system into a continuous optimization framework.

1.2. A Paradigm Shift: From Temporal Logic to Geometric Reachability

This report proposes a fundamental departure from this gradient-centric paradigm. Inspired by the "Beyond Gradients" framework, which recasts SNN learning as causal inference on a

Temporal Event Hypergraph (TEH) , this work introduces a novel conceptual translation. The TEH model successfully replaces the numerical propagation of gradients with the discrete traversal of causal evidence, but its primitives remain rooted in time-stamped events and temporal logic. The framework presented herein argues for a further abstraction: replacing the discrete, temporal logic of the TEH model with a continuous, geometric logic operating within a constrained, perceptually meaningful vector space.

The central thesis is to shift the representation of neural computation from the temporal domain to a spectral one. Each neuron's observable state at a given moment is no longer a binary spike or a continuous potential, but a *color*—a point in a three-dimensional vector space. Causal dependencies, previously modeled as hyperedges connecting events in time, are re-envisioned as geometric mixing operations that combine these colors. Consequently, the core question of credit assignment—"Which past events caused this outcome?"—is transformed. The check for "causal admissibility" in a hypergraph, which relies on satisfying temporal constraints, is replaced by a check for "geometric reachability" within a physically constrained volume. This move from a discrete graph traversal to a continuous geometric query allows for a new, powerful, and natively gradient-free approach to learning in SNNs.

1.3. Thesis Overview: Causality within the OKLab Perceptual Droplet

The computational substrate for this new model is the OKLab color space, a modern, perceptually uniform space designed by Björn Ottosson. A key property of OKLab is that the Euclidean distance between two color points corresponds directly to their perceived difference, a metric known as \Delta E. This property is not merely a convenience; it provides a natural and meaningful metric for defining error, tolerance, and optimization objectives within the model. A linear interpolation between two colors in OKLab space results in a perceptually smooth gradient, a feature notably absent in many other color spaces like CIELAB, where such operations can introduce unexpected hue shifts. By grounding the model's fundamental operation—linear mixing—in a space where it has well-behaved perceptual consequences, the learning dynamics become more stable and predictable.

Crucially, the set of all valid colors displayable within a standard gamut (e.g., sRGB) forms a specific, non-uniform 3D volume when represented in OKLab. This volume, which resembles a distorted droplet, is not an abstract mathematical construct but a hard physical constraint. This "gamut droplet" defines the boundaries of all possible valid neural states. A state is computationally feasible if and only if its corresponding color lies within this droplet. This physical boundary becomes the fundamental law of the system, and the core learning problem becomes one of navigating and reasoning about reachability within this constrained geometric world.

II. The OKLab Perceptual Droplet: A Geometric Substrate for Neural Computation

2.1. Formalizing the Computational Space

To build a computational model upon the OKLab space, it is necessary to first establish its formal mathematical properties. The OKLab space is a three-dimensional real vector space, where each point corresponds to a unique color perception under standard viewing conditions.

A color, denoted by a vector x, is represented in Cartesian coordinates as $x = (L, a, b) \in \mathbb{R}^3$. These components have specific perceptual interpretations :

- L: The perceptual lightness, normalized to the range $$, where $L=0$ corresponds to pure black and $L=1$ corresponds to a reference white (D65).
- a: The green-red opponent axis. Negative values of a correspond to green hues, while positive values correspond to red hues.
- b: The blue-yellow opponent axis. Negative values of b correspond to blue hues, while positive values correspond to yellow hues.

For many applications, including the analysis of color saturation and hue, it is more intuitive to work in a cylindrical coordinate system, denoted OKLch. A color x can be represented as a tuple (L, C, h), where L is the same lightness component, and:

- C: The chroma, representing the color's intensity or saturation. It is calculated as the Euclidean distance from the neutral (gray) axis: $C = \sqrt{a^2 + b^2}$. $C=0$ for any achromatic color (black, gray, white).
- h: The hue angle, representing the pure color direction. It is calculated as $h = \operatorname{atan2}(b, a)$, typically expressed in degrees or radians.

The Cartesian coordinates can be recovered from the cylindrical form via the transformations $a = C \cos(h)$ and $b = C \sin(h)$. A foundational property of the OKLab space is its perceptual uniformity. The perceived difference between two colors, x_1 and x_2, is quantified by the ΔE metric, which is defined simply as the Euclidean distance between their vector representations in the (L, a, b) space : $\Delta E(x_1, x_2) = \|x_1 - x_2\|_2 = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}$ This property provides an intrinsic and perceptually meaningful metric for defining loss functions and tolerances within the learning framework. A ΔE value of approximately 1.0 to 2.0 is considered the threshold of just-noticeable difference for the human eye.

2.2. The Geometry of Feasibility: The Gamut Droplet $\mathcal{D}$

While the OKLab space itself is an unbounded vector space, the set of colors that can be physically realized by a given device, such as a standard sRGB monitor, is a finite subset. When the sRGB gamut is mapped into the OKLab space, it forms a complex, non-convex 3D solid, hereafter referred to as the gamut droplet, $\mathcal{D}$. This droplet is not a simple geometric primitive like a cube or sphere; it is widest at mid-lightness values and tapers significantly towards pure black ($L=0$) and pure white ($L=1$), reflecting the fact that highly saturated colors can only be produced at intermediate brightness levels.

The geometry of this droplet can be precisely characterized. For any given hue angle h, a 2D cross-section of the droplet in the (L, C) plane reveals a shape resembling a deformed triangle. The boundary of this slice is defined by two segments: a straight line extending from pure black ($L=0$, $C=0$) to a point of maximum chroma, known as the "cusp," and a curved line segment from the cusp to pure white ($L=1$, $C=0$). The location of the cusp and the shape of the curved boundary are functions of the hue angle h.

This complex boundary defines the fundamental law of computational feasibility in our framework. A neural state, represented by a color x, is valid if and only if it lies within or on the surface of this droplet. This condition can be formalized by defining a "gamut oracle" function, $C_{\max}(L, h)$, which returns the maximum possible chroma for any given lightness L and hue h. The gamut droplet $\mathcal{D}$ can then be formally defined as the set:

$$\mathcal{D} = \{x = (L, a, b) \in \mathbb{R}^3 \mid \sqrt{a^2 + b^2} \le C_{\max}(L, \operatorname{atan2}(b, a))\}$$

In a practical implementation, the function $C_{\max}(L, h)$ can be realized as a pre-computed 2D lookup table over a grid of (L, h) values, with bilinear or trilinear interpolation used to query values between grid points. This oracle is the arbiter of physical possibility for any color generated by the network.

2.3. The Inclined Double-Cone Parametrization: In-Gamut by Construction

A direct representation of a neuron's state in (L, C, h) coordinates presents a challenge: any arbitrary modification to these parameters could easily produce a color that lies outside the gamut droplet $\mathcal{D}$, resulting in an invalid state. To address this, we introduce a novel parametrization, inspired by the geometry of an inclined double cone , that guarantees any generated state is valid by construction.

Instead of representing a neuron's state directly by its chroma C, we introduce a latent, normalized saturation parameter, $S' \in $. The state of a neuron is thus parameterized by the tuple (L, S', h). The true, physical chroma C is then *derived* from this latent representation using the gamut oracle: $C = S' \cdot C_{\max}(L, h)$ The final Cartesian coordinates (L, a, b) are then computed as usual: $a = C \cos(h)$ and $b = C \sin(h)$. This construction ensures that the generated chroma C can never exceed the maximum possible chroma $C_{\max}$ for its given lightness and hue. Consequently, any color generated from the (L, S', h) parametrization is guaranteed to be in-gamut. This "safe" representation prevents the generation of invalid states and simplifies all downstream computations by ensuring that every input to the causal mixing law is a physically realizable color.

This parametrization also provides a canonical path for modifying a color's saturation. By holding L and h constant and varying S' from 1 down to 0, the color traces a straight line in (L,a,b) space from its position on the gamut boundary towards the neutral gray point (L, 0, 0) on the central axis. The latent parameter S' can be interpreted as a *perceptually normalized* saturation. A uniform change in S' will produce a large change in absolute chroma C in regions where the gamut is wide (i.e., $C_{\max}$ is large) and a small change in C where the gamut is narrow. This mechanism provides a form of built-in adaptive gain control, allowing the network to learn to manipulate a uniform latent variable whose real-world effect is automatically scaled by the local geometric properties of the perceptual space.

III. A Color-Geometric Model of Causal Propagation

3.1. The Causal Mixing Law

The core computational primitive of the framework is the synthesis of a postsynaptic neuron's state from the states of its presynaptic inputs. This operation is designed as a direct geometric analogue to the n-ary hyperedge in the TEH model. Instead of a logical aggregation of temporal events, we define a physical law of additive color mixing.

For a postsynaptic neuron v with a set of presynaptic neurons $\mathcal{P}(v)$, its state x_v is computed as a convex combination of the presynaptic states $\{x_u\}_{u \in \mathcal{P}(v)}$ and an optional bias color, x_{bias}. The mixing law is formally defined as: $x_v = \sum_{u \in \mathcal{P}(v)} \alpha_{vu} x_u + \beta_v x_{\text{bias}}$ subject to the constraints: $\alpha_{vu} \ge 0, \quad \beta_v \ge 0, \quad \text{and} \quad \sum_{u \in \mathcal{P}(v)} \alpha_{vu} + \beta_v = 1$

The coefficients $\{\alpha_{vu}\}$ and β_v are the synaptic weights, representing the proportional contribution of each input. Geometrically, this operation places the resulting color x_v within the convex hull of the presynaptic colors and the bias color. The bias term x_{bias} acts as a crucial anchor, providing the neuron with additional degrees of freedom. By setting x_{bias} to neutral gray, D65 white, or black, the neuron can learn to lighten, darken, or desaturate the combined input signal, enabling a richer computational palette.

3.2. Causality as Geometric Reachability

With the forward computational step defined, we can now establish the central principle of causality within the framework. A causal link is deemed valid not by satisfying temporal logic, but by demonstrating geometric reachability.

The **Local Causal Condition** states that a set of presynaptic colors $\{x_u\}$ is a valid cause for an observed postsynaptic color x_v^{obs} if and only if x_v^{obs} can be expressed as a convex combination of the presynaptic colors (and bias) *and* this combination lies within the gamut droplet $\mathcal{D}$. This dual constraint defines the set of all locally reachable states. Formally, a causal explanation exists if:

$$x_v^{\text{obs}} \in \text{conv}(\{x_u\}_{u \in \mathcal{P}(v)} \cup \{x_{\text{bias}}\}) \cap \mathcal{D}$$

This condition gives rise to two distinct modes of causal failure:

1. **Directional Insufficiency:** If x_v^{obs} lies outside the convex hull of the inputs, the presynaptic palette is fundamentally incapable of producing the target color, regardless of gamut constraints. The inputs lack the necessary hue or lightness characteristics.
2. **Chromatic Impossibility:** If x_v^{obs} lies inside the convex hull but outside the gamut droplet $\mathcal{D}$, the target is too saturated (has too much chroma) to be physically realized from the given inputs, even though its hue and lightness are directionally achievable.

3.3. Multi-Step Causal Propagation and Reachable Sets

The local causal condition can be extended to describe the propagation of information through multiple layers of the network or across time steps. This is achieved by tracking the evolution of "reachable sets"—the volumes in OKLab space that each neuron can possibly occupy.

Let $\mathcal{R}_t(u)$ be the set of all reachable colors for a neuron u at layer or time step t. The reachable set for a postsynaptic neuron v at the next step, $\mathcal{R}_{t+1}(v)$, is defined as the intersection of the gamut droplet $\mathcal{D}$ with the convex hull of the union of the reachable sets of all its presynaptic neurons. This operation, known as the convex hull sum or Minkowski sum of the individual convex hulls, is then clipped by the gamut boundary:

$$\mathcal{R}_{t+1}(v) = \left(\text{conv}\left(\bigcup_{u \in \mathcal{P}(v)} \mathcal{R}_t(u) \cup \{x_{\text{bias}}\} \right) \right) \cap \mathcal{D}$$

An observed state $x_v^{\text{obs}}(t+1)$ is considered causally explainable by the network's history up to time t if and only if $x_v^{\text{obs}}(t+1) \in \mathcal{R}_{t+1}(v)$.

This propagation mechanism reveals a fundamental dynamic of the system. The convex hull operation is inherently contractive; it averages colors, pulling them away from the extremities of the input sets and towards the neutral center. The intersection with $\mathcal{D}$ can further shrink the reachable volume. This implies that as information propagates through the network, the reachable sets $\mathcal{R}_t$ have a natural tendency to become smaller and less chromatically expansive, collapsing towards the gray axis. For the network to maintain vibrant, high-chroma representations in its deeper layers, it must learn to counteract this entropic pull. The learning algorithm must discover highly specific, non-uniform weightings (α_{vu}) that selectively amplify certain inputs, actively "pulling" the resulting mixture back out towards the gamut boundary. This creates a dynamic tension between the contractive nature of mixing and the expressive requirements of representation, a core computational challenge that learning must solve.

IV. Credit Assignment as Geometric Decomposition

4.1. The Geometric Preimage Problem

In a learning context, credit assignment is the process of determining which presynaptic inputs are responsible for a given postsynaptic outcome, particularly when there is a mismatch between the actual and desired state. In this framework, credit assignment is reframed as a geometric inverse problem: the search for a "geometric preimage." This process is the direct analogue of the backward hyperpath traversal in the TEH model.

The problem can be stated as follows: Given a desired target color for a postsynaptic neuron, x_v^*, and the set of observed colors from its presynaptic inputs, $\{x_u\}$, find the non-negative mixing coefficients, $\{\alpha_{vu}\}$, that best reconstruct x_v^* while satisfying all physical and structural constraints. The solution to this problem, the set of optimal coefficients α^*, directly represents the credit or blame assigned to each presynaptic connection.

4.2. Formulation as Constrained Non-Negative Least Squares (NNLS)

The geometric preimage problem can be formalized as a constrained optimization task. The objective is to find the mixing coefficients that minimize the perceptual error between the synthesized color and the target color. Given the definition of ΔE in OKLab, this is equivalent to minimizing the squared Euclidean distance. This formulation is a Non-Negative Least Squares (NNLS) problem with additional constraints.

Let X be a $3 \times N$ matrix where each column is the (L, a, b) vector of a presynaptic color x_u, and let α be the $N \times 1$ vector of unknown coefficients α_{vu}. The optimization problem is:

$$\min_{\alpha \ge 0, \mathbf{1}^\top\alpha \le 1} \quad \| X\alpha + (1 - \mathbf{1}^\top\alpha)x_{\text{bias}} - x_v^* \|_2^2$$

subject to the additional gamut feasibility constraint:

$$X\alpha + (1 - \mathbf{1}^\top\alpha)x_{\text{bias}} \in \mathcal{D}$$

This is a Quadratic Programming (QP) problem, as it involves minimizing a quadratic objective function subject to linear inequality constraints (non-negativity and the simplex constraint $\sum \alpha_{vu} + \beta_v = 1$) and a non-linear convex-set constraint (in-gamut feasibility). This type of problem can be solved efficiently using standard numerical optimization libraries, such as scipy.optimize.minimize with the 'SLSQP' (Sequential Least Squares Programming) method, which is designed to handle such mixed constraints. The solution vector α^* provides the barycentric coordinates of the projection of x_v^* onto the reachable set, effectively decomposing the target into its constituent causal sources.

4.3. Modeling Dendritic Coincidence via k-Sparsity

The basic NNLS formulation assumes that all presynaptic neurons can potentially contribute to

the postsynaptic state. However, biological neurons often operate on the principle of coincidence detection, firing only in response to a small coalition of near-simultaneous inputs. This n-ary dependency, modeled as B-connectivity in the TEH framework, can be captured geometrically by introducing a sparsity constraint.
We augment the NNLS problem with the constraint that the solution vector α must be k-sparse, meaning it can have at most k non-zero elements: $\|\alpha\|_0 \le k$ where k is a small integer, typically in the range of 2 to 5. This constraint enforces that the target color must be explained by a "minimal color coalition," reflecting the need for a small group of presynaptic inputs to coincide.
This transforms the credit assignment task into a sparse regression problem. While enforcing an L_0-norm constraint is NP-hard in general, for the small number of inputs to a single neuron, it is computationally tractable. It can be solved using mixed-integer quadratic programming, approximated using L_1 regularization (Lasso) , or addressed with heuristic algorithms like orthogonal matching pursuit or top-k selection methods.
The introduction of this sparsity constraint has a profound effect on the geometry of the problem. The local reachable set is no longer a single, continuous convex polytope formed by all presynaptic inputs. Instead, it becomes a finite union of many smaller, lower-dimensional simplices, where each simplex is the convex hull of a specific k-sized coalition of inputs. The learning problem is thus transformed from navigating a single convex space to selecting the correct low-dimensional "facet" (i.e., the correct coalition) capable of reaching the target. This more structured, combinatorial landscape is a much closer geometric analogue to the discrete, logical nature of the original hypergraph model.

4.4. Gamut-Aware Scoring and Gradient-Free Updates

Once the optimal mixing coefficients α^* are found, a final score is computed for each synapse to modulate the weight update. This score incorporates not only the magnitude of the coefficient but also the robustness of the solution with respect to the gamut boundary. A solution that lies deep within the droplet is considered more robust and reliable than one that lies perilously close to the boundary.
We define a **gamut margin** factor that quantifies this robustness. For an optimal mix x_{mix}^* with coordinates (L^*, C^*, h^*), the margin is: $\text{margin} = \phi\left(1 - \frac{C^*}{C_{\max}(L^*, h^*)}\right)$ where ϕ is a monotonically increasing function (e.g., $\phi(z) = z$ or $\phi(z) = z^2$) that amplifies the score for solutions with a large margin (i.e., $C^* \ll C_{\max}$).
The final **evidence score** s_{vu} for the synapse from u to v is the product of its contribution and the solution's robustness: $s_{vu} = \alpha_{vu}^* \cdot \text{margin}$ This score is then used to drive a simple, gradient-free weight update rule. The direction of the update is determined by the sign of the error, while the magnitude is provided by the geometrically derived evidence score:
$$\Delta w_{vu} \propto \operatorname{sgn}\left(-\frac{\partial\mathcal{L}}{\partial x_v}\right) \cdot s_{vu}$$
Here, $\mathcal{L}$ is the task-level loss, and the sign term indicates whether the postsynaptic color x_v needs to be adjusted towards or away from a global target. This mechanism assigns credit based on a verifiable geometric decomposition, entirely bypassing the need for derivatives and backpropagation.

V. Algorithmic Blueprint and Practical Implementation

5.1. The Causal Update Algorithm (Pseudocode)

The concepts developed in the preceding sections can be synthesized into a practical, event-driven algorithm for performing a single learning update at a postsynaptic neuron v.

Inputs:

- A set of candidate presynaptic colors, $\{x_u\}$.
- A target postsynaptic color, x_v^*.
- An efficient implementation of the gamut oracle, $C_{\max}(L, h)$.
- Hyperparameters: sparsity level k, perceptual tolerance ϵ, learning rate η.

Procedure:

1. **Parameterize Safely:** Ensure all candidate and target colors are represented using the inclined double-cone parametrization (L, S', h) to guarantee they are in-gamut by construction. Convert to Cartesian (L, a, b) for linear algebra operations.
2. **Formulate the Problem:**
   - Construct the matrix X, where each column is the (L, a, b) vector of a presynaptic color x_u. Include the bias color x_{bias} as an additional column if used.
   - Define the objective function to minimize $\|X\alpha - x_v^*\|_2^2$.
   - Define the constraints: $\alpha_i \ge 0$, $\sum \alpha_i = 1$, and $\|\alpha\|_0 \le k$.
3. **Geometric Attribution (Solve):**
   - Solve the constrained, k-sparse Non-Negative Least Squares (NNLS) problem to find the optimal coefficients α^*. This can be implemented using a suitable quadratic programming solver.
4. **Feasibility Check and Scoring:**
   - Compute the resulting mixture: $x_{\text{mix}} = X\alpha^*$.
   - Convert x_{mix} to OKLch coordinates $(L_{\text{mix}}, C_{\text{mix}}, h_{\text{mix}})$.
   - **If** $C_{\text{mix}} \le C_{\max}(L_{\text{mix}}, h_{\text{mix}})$ and $\Delta E(x_{\text{mix}}, x_v^*) \le \epsilon$:
     - The target is reachable and the solution is valid.
     - Calculate the gamut margin: margin = $1 - (C_{mix} / C_{max}(L_{mix}, h_{mix}))$.
     - Calculate the evidence score for each synapse $u \to v$: $s_{vu} = \alpha_{vu}^* \cdot \text{margin}$.
     - Apply the weight update: $\Delta w_{vu} = \eta \cdot \operatorname{sgn}(\text{error}_v) \cdot s_{vu}$.
   - **Else:**
     - The target is unreachable given the constraints. Trigger an escalation procedure (see Section 5.2).

5.2. Strategies for Unreachability

A critical aspect of the framework is its ability to handle cases where the geometric attribution problem is infeasible—that is, when a target color x_v^* cannot be causally explained by the available presynaptic inputs. This situation requires a strategy for either adapting the target or propagating the error further back into the network.

1. **Target Projection:** The most direct approach is to accept the limitations of the local

causal neighborhood. The optimization solver can be tasked with finding the point in the reachable set ($\text{conv}(\{x_u\}) \cap \mathcal{D}$) that is closest to the unreachable target x_v^*. This closest point, x_v^{proj}, becomes the new, achievable target for the learning update. This strategy effectively attributes the discrepancy to "lossy synaptic propagation" and trains the network to get as close as possible.

2. **Basis Augmentation:** Infeasibility often arises because the convex hull of the presynaptic colors is too small. The local reachable set can be expanded by augmenting the basis of colors used in the mixture. This can involve introducing additional learned "basis neurons" that provide a richer palette of colors, or by dynamically increasing the weight given to static anchor colors like black, white, or neutral gray. This allows the network to learn to expand its local expressive capacity when needed.
3. **Backtracking and Escalation:** If a target x_v^* is fundamentally unreachable, the responsibility for the error must be assigned to the neurons that produced the insufficient presynaptic colors $\{x_u\}$. This is the geometric analogue of traversing one level deeper into the causal hypergraph. The unreachable target x_v^* is used to define new error signals for the preceding layer. Each presynaptic color x_u becomes a variable to be adjusted, and new target colors $\{x_u^*\}$ are computed such that their convex hull *would* contain x_v^*. These new targets $\{x_u^*\}$ are then used to drive the credit assignment process at the previous layer, effectively propagating the error one step backward through the geometric chain of causality.

5.3. Implementation Details and Computational Complexity

The practical viability of this framework hinges on several key implementation details.

- **Gamut Oracle:** The performance of the entire system depends on an efficient implementation of the $C_{\max}(L, h)$ function. A pre-computed lookup table on a sufficiently fine grid of (L, h) values, combined with fast bilinear or bicubic interpolation, is the most practical approach. This table can be computed once and stored, and values can be cached during training to minimize redundant lookups.
- **Perceptual Tolerance (ϵ):** The tolerance for matching the target color should be set in perceptually meaningful ΔE units. A starting value in the range of 1.0 to 2.0, corresponding to the threshold of human perception, is appropriate. This value can be gradually annealed to a smaller number during training to encourage higher precision.
- **Computational Complexity:** The framework's computational cost is fundamentally different from that of BPTT. Instead of dense matrix multiplications across all neurons and time steps, computation is sparse and event-driven. The primary computational load is a small quadratic program that must be solved for each postsynaptic event (or supervisory signal). For a neuron with N inputs and a sparsity constraint of k, the complexity of this solve is low and largely independent of the network's depth or temporal history. This event-driven, localized computation makes the framework exceptionally well-suited for parallel and asynchronous execution on neuromorphic hardware.

VI. Comparative Analysis: A Geometric Reinterpretation of the Hypergraph Framework

6.1. Establishing the Conceptual Isomorphism

The Color-Geometry framework was designed as a direct conceptual translation of the Temporal Event Hypergraph (TEH) model. Its core value proposition lies in preserving the *intent* of the TEH—a robust, interpretable, and discrete form of credit assignment—while shifting the underlying mathematical primitives from temporal logic and graph theory to continuous geometry and convex optimization. This section makes this isomorphism explicit, demonstrating how each key concept in the TEH framework has a direct and powerful analogue in the geometric model.

6.2. Table of Analogues

The following table provides a point-by-point comparison, mapping the foundational concepts of the TEH framework to their counterparts in the Color-Geometry framework. This mapping highlights the shared structural and philosophical underpinnings of the two approaches, fulfilling the central directive to mirror the original model's structure and intent.

Hypergraph Framework Concept	**Color-Geometry Framework Analogue**	**Mathematical Formalism**	**Interpretation & Nuances**
Causality Primitive: A discrete spike event (neuron_id, t) is a vertex.	**State Primitive:** A neuron's state is a color $x = (L, a, b)$ within the gamut droplet $\mathcal{D}$.	$x \in \mathcal{D} \subset \mathbb{R}^3$	The state is no longer just an event in time, but a point in a continuous, perceptually meaningful vector space.
Causal Link: A directed hyperedge (tails) → head represents a successful firing event.	**Causal Synthesis:** A convex mixture of presynaptic colors produces a postsynaptic color.	$x_v = \sum \alpha_{vu} x_u$	The causal link is not a discrete graph edge but a continuous mixing operation governed by the laws of additive light.
N-ary Dependency: B-Connectivity requires *all* presynaptic spikes in the tail to be present.	**Coincidence Detection:** A k-sparse coalition of presynaptic colors is sufficient to span the target.	$\lVert\alpha\rVert_0 \le$	The strict "all-or-nothing" logical AND is relaxed to a more flexible geometric "spanning" condition by a minimal subset.
Causality Check: Is a hyperedge valid based on temporal logic and neuron state?	**Reachability Check:** Is the target color within the convex hull of sources and inside the gamut droplet?	$x_v \in \text{conv}(\{x_u\}) \cap \mathcal{D}$	The check shifts from satisfying abstract temporal rules to satisfying concrete geometric and physical constraints.
Credit Assignment: Backward traversal of hyperpaths to find causal chains.	**Credit Attribution:** Solving a constrained NNLS problem to find the barycentric coordinates (mixing	$\alpha^* = \arg\min \lVert X\alpha - x^*\rVert_2$ s.t. constraints	Credit is found not by pathfinding, but by geometric decomposition. The solution vector α^*

Hypergraph Framework Concept	Color-Geometry Framework Analogue	Mathematical Formalism	Interpretation & Nuances
	weights).		*is* the credit distribution.
Evidence Aggregation: Scoring paths and summing scores to update weights.	**Evidence Scoring:** Weighting the solution α^* by the gamut margin to reward robust solutions.	$s_{vu} = \alpha_{vu}^* \cdot \text{margin}$	Credit is modulated by the "physical robustness" of the causal explanation; solutions near the boundary are brittle and less trusted.
Frame Invariance: The causal structure is independent of the specific temporal logic formalism used.	**Representation Invariance:** The reachability decision (in vs. out) is a geometric fact, independent of the coordinate system (Lab, Lch).	$\text{conv}(S) \cap \mathcal{D}$ is invariant under isometries.	The fundamental causal verdict is absolute, reinforcing the physical analogy and the objectivity of the computation.

VII. Conclusion and Future Directions

7.1. Summary of the Framework

This report has detailed a novel paradigm for learning in Spiking Neural Networks, reframing the problem of credit assignment from one of gradient-based optimization to one of causal inference within a constrained geometric space. By representing neural states as colors in the perceptually uniform OKLab space and modeling causal dependencies as convex mixing operations, the Color-Geometry Causal Framework offers a complete, gradient-free learning mechanism. Its core principles—causality as geometric reachability and credit assignment as geometric decomposition—provide a robust and interpretable alternative to the fragile and opaque methods based on backpropagation. The framework is computationally aligned with the event-driven, sparse nature of SNNs, replacing the dense overhead of BPTT with localized, small-scale quadratic programs. The hard physical constraint of the gamut droplet provides a natural and powerful form of regularization, grounding the entire system in the realm of physically realizable states.

7.2. Bridging Subsymbolic and Symbolic AI

A key advantage inherited from its conceptual predecessor, the TEH model, is a profound shift towards interpretability. Traditional neural networks encode knowledge implicitly in their weight matrices, creating opaque "black box" systems. In the geometric framework, every learning update is the result of a verifiable causal proof. The solution to the NNLS problem, the vector α^*, is not an abstract gradient but a concrete decomposition of the target state into its constituent sources. This, combined with the gamut margin score, provides a structured, machine-readable explanation for why a particular synapse was strengthened or weakened. This intrinsic "glass box" nature is a significant step towards building trust and enabling formal verification in complex neural systems, bridging the gap between the subsymbolic processing of neural networks and the auditable reasoning of symbolic AI.

7.3. Future Research Directions

The introduction of this geometric paradigm opens several promising avenues for future research:

- **Exploring Alternative Geometries:** While OKLab provides an excellent substrate due to its perceptual uniformity, other advanced color spaces such as JzAzBz or CAM16-UCS offer different geometric properties and trade-offs between perceptual accuracy and computational simplicity. A systematic investigation into how the unique geometry of each space's gamut droplet affects learning dynamics and representational capacity is a critical next step.
- **Dynamic Basis and Learned Primitives:** The current framework utilizes a static bias color. Future work should explore methods that allow the network to learn its own set of "basis colors." This would enable the network to dynamically construct an optimal, task-specific color palette, potentially increasing its expressive power and learning efficiency.
- **Hardware Co-Design:** The computational primitives of this framework—small, parallel NNLS/QP solves and fast table lookups for the gamut oracle—are distinct from the multiply-accumulate operations that dominate conventional deep learning. This invites research into co-designing neuromorphic accelerators with dedicated hardware units optimized for these geometric operations, fully leveraging the algorithm's event-driven nature and inherent parallelism.
- **Theoretical Foundations:** The propagation of reachable sets, \mathcal{R}_t, through the network represents a transformation of information. There are deep potential connections between the geometric evolution of these sets and concepts from information theory, such as the Data Processing Inequality. Formalizing the notion of "chromatic information" and analyzing how it is preserved, compressed, or lost at each layer could provide a rigorous theoretical foundation for understanding the computational properties of this new class of neuro-geometric systems.

Works cited

1. A perceptual color space for image processing - Björn Ottosson, https://bottosson.github.io/posts/oklab/ 2. Björn Ottosson, https://bottosson.github.io/ 3. Interview With Björn Ottosson, Creator Of The Oklab Color Space - Smashing Magazine, https://www.smashingmagazine.com/2024/10/interview-bjorn-ottosson-creator-oklab-color-space/ 4. Oklab color space - Wikipedia, https://en.wikipedia.org/wiki/Oklab_color_space 5. Color Distance and Delta E - ColorAide Documentation, https://facelessuser.github.io/coloraide/distance/ 6. Porting OkLab colorspace to integer arithmetic - ubitux/blog, http://blog.pkh.me/p/38-porting-oklab-colorspace-to-integer-arithmetic.html 7. Okay, Color Spaces — ericportis.com, https://ericportis.com/posts/2024/okay-color-spaces/ 8. Studying Gamut Clipping - Simon's Tech Blog, http://simonstechblog.blogspot.com/2021/05/studying-gamut-clipping.html 9. What Is Delta E? And Why Is It Important for Color Accuracy? - ViewSonic Library, https://www.viewsonic.com/library/creative-work/what-is-delta-e-and-why-is-it-important-for-color-accuracy/ 10. 4.4 Lab Colour Space and Delta E Measurements, https://opentextbc.ca/graphicdesign/chapter/4-4-lab-colour-space-and-delta-e-measurements/

11. OKLab / OKLch palette rotation / Gabe Coulter - Observable, https://observablehq.com/@coulterg/oklab-oklch-palette-rotation 12. More experiments with sRGB gamut boundary in L*a*b* space - MATLAB Central Blogs, https://blogs.mathworks.com/steve/2022/05/03/more-experiments-with-srgb-gamut-boundary-in-lab-space/ 13. The double cone - Math Insight, https://mathinsight.org/double_cone 14. Conic Sections: The Double Cone | Wolfram Demonstrations Project, https://demonstrations.wolfram.com/ConicSectionsTheDoubleCone/ 15. Convex hull - Wikipedia, https://en.wikipedia.org/wiki/Convex_hull 16. Chapter 3 Convex Hull, https://ti.inf.ethz.ch/ew/courses/CG13/lecture/Chapter%203.pdf 17. Computing the Convex Hull of Line Intersections - Purdue e-Pubs, https://docs.lib.purdue.edu/context/cstech/article/1418/viewcontent/ocr.TR_2084_499.pdf 18. python - How to include constraint to Scipy NNLS function solution ..., https://stackoverflow.com/questions/33385898/how-to-include-constraint-to-scipy-nnls-function-solution-so-that-it-sums-to-1 19. Non-negative least squares — scikit-learn 1.4.2 documentation, https://scikit-learn.org/1.4/auto_examples/linear_model/plot_nnls.html 20. nnls — SciPy v1.16.2 Manual, https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.nnls.html 21. Quadratic programming - Wikipedia, https://en.wikipedia.org/wiki/Quadratic_programming 22. The Simplex Method for Quadratic Programming: Notes on Linear Programming and Extensions-Part 51 - RAND, https://www.rand.org/content/dam/rand/pubs/research_memoranda/2009/RM2388.pdf 23. Algorithms for projecting a point onto the convex hull spanned by a set of vectors, https://math.stackexchange.com/questions/3109906/algorithms-for-projecting-a-point-onto-the-convex-hull-spanned-by-a-set-of-vecto 24. Barycentric coordinate system - Wikipedia, https://en.wikipedia.org/wiki/Barycentric_coordinate_system 25. 1.1. Linear Models — scikit-learn 1.7.2 documentation, https://scikit-learn.org/stable/modules/linear_model.html 26. Least square regression with L1 regularization and non-negativity constraint, https://stats.stackexchange.com/questions/87559/least-square-regression-with-l1-regularization-and-non-negativity-constraint 27. Top-KAST: Top-K Always Sparse Training - NIPS, https://proceedings.neurips.cc/paper/2020/file/ee76626ee11ada502d5dbf1fb5aae4d2-Paper.pdf 28. OKRidge: Scalable Optimal k-Sparse Ridge Regression, https://proceedings.neurips.cc/paper_files/paper/2023/file/80f48ffa8022773973a4a5cec7cce19c-Paper-Conference.pdf 29. Scaling and evaluating sparse autoencoders | OpenAI, https://cdn.openai.com/papers/sparse-autoencoders.pdf