GRADED NEURAL NETWORKS

TONY SHASKA

To my parents, brilliant teachers dismissed for their beliefs in a system where grading meant obedience.

Abstract. We introduce a rigorous framework for *Graded Neural Networks*, a new class of architectures built on coordinate-wise graded vector spaces $\mathcal{V}_{\mathbf{q}}^n$. Using an algebraic scalar action $\lambda \star \mathbf{x} = (\lambda^{q_i} x_i)$ defined by a grading tuple $\mathbf{q} = (q_0, \ldots, q_{n-1})$, we construct grade-sensitive neurons, activations, and loss functions that embed hierarchical feature structure directly into the network's architecture. This grading endows GNNs with enhanced expressivity and interpretability, extending classical neural networks as a special case.

We develop both the algebraic theory and computational implementation of GNNs, addressing challenges such as numerical instability and optimization with anisotropic scaling. Theoretical results establish universal approximation for graded-homogeneous functions, along with convergence rates in graded Sobolev and Besov spaces. We also show that GNNs achieve lower complexity for approximating structured functions compared to standard networks.

Applications span hierarchical data modeling, quantum systems, and photonic hardware, where grades correspond to physical parameters. This work provides a principled foundation for incorporating grading into neural computation, unifying algebraic structure with learning, and opening new directions in both theory and practice.

1. Introduction

Many real-world datasets exhibit hierarchical, heterogeneous, or structured features that standard neural networks treat uniformly. However, in fields ranging from algebraic geometry and quantum physics to temporal signal processing and photonic computing, inputs vary in importance or scale, suggesting the need for architectures that respect this asymmetry. This paper introduces a generalized neural framework—*Graded Neural Networks* (GNNs)—that models such data over coordinate-wise graded vector spaces $\mathcal{V}_{\mathbf{q}}^n$, using algebraic structure to inform both architecture and learning.

This work builds on the foundational model of Artificial Neural Networks over graded spaces proposed in [1]. We extend that paradigm by incorporating multiplicative neurons, exponential activations, and robust loss functions, including a general homogeneous loss tailored to graded structures, while addressing numerical instability and optimization challenges in high-dimensional settings. The result is a flexible and expressive architecture that generalizes classical neural networks when $\mathbf{q} = (1, \ldots, 1)$, with graded ReLU ($\max\{0, |x_i|^{1/q_i}\}$) outputting positive values for negative inputs, enhancing feature scaling but potentially reducing sparsity.

Key words and phrases. Graded Neural Networks, Graded Vector Spaces.

Our motivation stems from algebraic geometry: the moduli space of genus two curves, embedded in the weighted projective space $\mathbb{P}_{(2,4,6,10)}$, uses graded invariants (J_2, J_4, J_6, J_{10}) to reflect geometric structure. In [2], neural networks trained on ungraded coefficients achieved only 40% accuracy in predicting automorphism groups or (n, n)-split Jacobians, whereas graded inputs boosted performance to 99%. This result, driven by $\mathbb{P}_{(2,4,6,10)}$'s parametrization of isomorphism classes, raises a broader question: does embedding algebraic grading into the network architecture systematically improve performance?

Similar grading arises in quantum systems, where supersymmetry distinguishes bosonic and fermionic components ($\mathbf{q} = (2, 1)$); in temporal signal processing, where higher grades prioritize recent inputs ($\mathbf{q} = (1, 2, 3, \ldots)$); and in photonic computing, where grades map to physical parameters like laser wavelengths. These applications motivate a general framework for neural networks that incorporates grading into representation and computation.

In Section 2, we introduce graded vector spaces, generalizing $\mathbb{R}^n$ through a decomposition into subspaces indexed by a set I (e.g., $\mathbb{N}$, $\mathbb{Q}$), with a scalar action $\lambda \star \mathbf{x} = (\lambda^{q_i} x_i)$ for coordinate-wise spaces $\mathcal{V}^n_{\mathbf{q}}(k)$. We define graded linear maps preserving this structure, prove their properties (Prop. 2), and introduce operations like direct sums and tensor products, with grades like $q_i + r_j$ for tensor bases. Norms (Euclidean, graded Euclidean, homogeneous, max-graded) are defined, with convexity properties analyzed, and a graded/filtered equivalence lemma connects grading to hierarchical representations, enabling coarse-to-fine learning in GNNs [1].

In Section 3, we define Graded Neural Networks (GNNs) over coordinate-wise graded vector spaces $\mathcal{V}^n_{\mathbf{q}}(k)$, with scalar action $\lambda \star \mathbf{x} = (\lambda^{q_i} x_i)$, introducing additive $(\sum w_i^{q_i} x_i + b)$ and multiplicative $(\prod (w_i x_i)^{q_i} + b)$ neurons, graded ReLU $(\max\{0, |x_i|^{1/q_i}\})$, exponential activations, and loss functions like graded MSE, Huber, and homogeneous loss, tailored to hierarchical data [1]. We prove convexity of the graded norm loss (Lem. 2), exact representation of graded-homogeneous polynomials (Thm. 7), Lipschitz continuity of activations (Thm. 8), and convergence of grade-adaptive optimization with learning rates $\eta_i = \eta/q_i$ (Thm. 9), with GNNs reducing to classical networks when $\mathbf{q} = (1, \ldots, 1)$ (Thm. 6).

In Section 4, we introduce the computational implementation, approximation capabilities, and applications of GNNs over $\mathcal{V}^n_{\mathbf{q}}$, addressing numerical challenges from the scalar action $\lambda \star \mathbf{x} = (\lambda^{q_i} x_i)$ using logarithmic stabilization and sparse matrix techniques to reduce complexity from $O(n^2)$ to $O(\sum_{j \in I_l} d_{l,j} d_{l-1,j})$ [1]. We prove universal approximation for graded-homogeneous functions (Thm. 11), exact representation of monomials (Prop. 4), and optimal convergence rates in graded Sobolev spaces (Thm. 13, achieving $O(m^{-k/n})$ for $f \in W^{k,2}_{\mathbf{q}}(K)$) and Besov spaces (Thm. 15, achieving $O(m^{-s/n})$ for $f \in B^s_{p,r,\mathbf{q}}(K)$), demonstrating superior efficiency over classical networks requiring more neurons for structured functions (Prop. 5, Cor. 2), with applications in machine learning for genus two invariants, quantum physics for supersymmetric systems, and neuromorphic computing.

The significance of GNNs lies in their ability to integrate algebraic structure with deep learning, offering a principled approach to modeling hierarchical data. By embedding the scalar action $\lambda \star \mathbf{x} = (\lambda^{q_i} x_i)$ into the architecture, GNNs align with the intrinsic geometry of graded vector spaces $\mathcal{V}^n_{\mathbf{q}}$, enhancing expressivity and interpretability. Theoretical results, including universal approximation (Thm. 11),

exact polynomial representation (Thm. 7), activation stability (Thm. 8), optimization convergence (Thm. 9), and efficiency over classical networks (Prop. 5, Cor. 2), provide a robust mathematical foundation, enabling reliable implementation across diverse domains.

Potential applications of GNNs extend beyond current examples to transformative use cases. In algebraic geometry, they model complex invariants like those of genus two curves ($\mathbf{q} = (2, 4, 6, 10)$) [2]; in quantum physics, they capture supersymmetric structures ($\mathbf{q} = (2, 1)$); and in temporal signal processing, they prioritize recent data ($\mathbf{q} = (1, 2, 3, \ldots)$). Emerging areas include neuromorphic computing, where grades align with synaptic weights, and graph-graded networks, where node/edge grading enhances graph-based learning, paving the way for advances in scientific computing and machine learning.

2. Graded Vector Spaces

Here we provide the essential background on graded vector spaces, extended to incorporate recent advancements in their structure and operations. The interested reader can check details at [3], [4], [5], [1], among other places. We use "grades" to denote the indices of grading (e.g., q_i), distinguishing them from "weights" used for neural network coefficients in Section 3.

A graded vector space is a vector space endowed with a grading structure, typically a decomposition into a direct sum of subspaces indexed by a set I. While we primarily focus on the traditional decomposition $V = \bigoplus_{n\in\mathbb{N}} V_n$ and the coordinate-wise form $\mathcal{V}^n_{\mathbf{q}}(k) = k^n$ with scalar action $\lambda\star\mathbf{x} = (\lambda^{q_i}x_i)$, these definitions generalize to arbitrary index sets I, including rational numbers, finite groups, or abstract algebraic structures, allowing greater flexibility in modeling hierarchical data; see [1]. These definitions are equivalent via basis choice, a perspective we adopt for neural networks in Section 3.

2.1. Generalized Gradation.

Let I be an index set, which may be $\mathbb{N}$, $\mathbb{Z}$, a field like $\mathbb{Q}$, or a monoid. An I-graded vector space V is a vector space with a decomposition:

$$V = \bigoplus_{i\in I} V_i,$$

where each V_i is a vector space, and elements of V_i are homogeneous of degree i. When $I = \mathbb{Q}$, grades can represent fractional weights, useful for modeling continuous hierarchies in machine learning tasks as in [1]. For $I = \mathbb{N}$, we recover the standard $\mathbb{N}$-graded vector space, often simply called a **graded vector space**.

Graded vector spaces are prevalent. For example, the set of polynomials in one or several variables forms a graded vector space, with homogeneous elements of degree n as linear combinations of monomials of degree n.

Example 1. *Let k be a field and consider $\mathcal{V}_{(2,3)}$, the space of homogeneous polynomials of degrees 2 and 3 in $k[x, y]$. It decomposes as $\mathcal{V}_{(2,3)} = V_2 \oplus V_3$, where V_2 is the space of binary quadratics and V_3 the space of binary cubics. For $\mathbf{u} = [f, g] \in V_2 \oplus V_3$, scalar multiplication is:*

$$\lambda \star \mathbf{u} = \lambda \star [f, g] = [\lambda^2 f, \lambda^3 g],$$

reflecting grades 2 and 3.

Next we will present an example that played a pivotal role in the invention of graded neural networks.

Example 2 (Moduli Space of Genus 2 Curves)**.** *Assume char $k \neq 2$ and C a genus 2 curve over k, with affine equation $y^2 = f(x)$, where $f(x)$ is a degree 6 polynomial. The isomorphism class of C is determined by its invariants J_2, J_4, J_6, J_{10}, homogeneous polynomials of grades 2, 4, 6, and 10, respectively, in the coefficients of C. The moduli space of genus 2 curves over k is isomorphic to the weighted (graded) projective space $\mathbb{P}_{(2,4,6,10),k}$.*

2.2. **Graded Linear Maps.** For an index set I, a linear map $f : V \to W$ between I-graded vector spaces is a **graded linear map** if it preserves the grading, $f(V_i) \subseteq W_i$, for all $i \in I$. Such maps are also called **homomorphisms (or morphisms) of graded vector spaces** or homogeneous linear maps. For a commutative monoid I (e.g., $\mathbb{N}$), maps homogeneous of degree $i \in I$ satisfy:

$$f(V_j) \subseteq W_{i+j}, \quad \text{for all } j \in I,$$

where $+$ is the monoid operation. If I is a group (e.g., $\mathbb{Z}$) or a field (e.g., $\mathbb{Q}$), maps of degree $i \in I$ follow similarly, with the operation defined by the structure of I; see [1]. A map of degree $-i$ satisfies:

$$f(V_{i+j}) \subseteq W_j, \quad f(V_j) = 0 \text{ if } j - i \notin I.$$

Proposition 1. *Let $\mathcal{V}_{\mathbf{q}}^n(k)$ and $\mathcal{V}_{\mathbf{q}'}^m(k)$ be graded vector spaces with grading vectors $\mathbf{q} = (q_0, \ldots, q_{n-1})$ and $\mathbf{q}' = (r_0, \ldots, r_{m-1})$, respectively, where $q_i, r_j \in \mathbb{Q}_{>0}$. Let $L : \mathcal{V}_{\mathbf{q}}^n(k) \to \mathcal{V}_{\mathbf{q}'}^m(k)$ be a k-linear map, and let $A = (a_{ij}) \in Mat_{m\times n}(k)$ be its matrix with respect to the standard bases.*

Then L is homogeneous of degree $d \in \mathbb{Q}$ if and only if

$$a_{ij} \neq 0 \quad \Longrightarrow \quad r_i = q_j + d.$$

*In particular, L is **grade-preserving** (i.e., $d = 0$) if and only if $a_{ij} \neq 0$ implies $r_i = q_j$.*

Proof. The scalar action on $\mathcal{V}_{\mathbf{q}}^n(k)$ is defined by $\lambda \star \mathbf{x} = (\lambda^{q_0} x_0, \ldots, \lambda^{q_{n-1}} x_{n-1})$, and similarly on $\mathcal{V}_{\mathbf{q}'}^m(k)$.

Suppose L is homogeneous of degree $d \in \mathbb{Q}$. Then for all $\lambda \in k^\times$ and all $\mathbf{x} \in \mathcal{V}_{\mathbf{q}}^n(k)$, we have:

$$L(\lambda \star \mathbf{x}) = \lambda^d \star L(\mathbf{x}),$$

which in coordinates becomes:

$$L\left(\lambda^{q_0} x_0, \ldots, \lambda^{q_{n-1}} x_{n-1}\right) = \left(\sum_{j=0}^{n-1} a_{0j} \lambda^{q_j} x_j, \ldots, \sum_{j=0}^{n-1} a_{m-1,j} \lambda^{q_j} x_j\right),$$

and on the other hand,

$$\lambda^d \star L(\mathbf{x}) = \left(\lambda^{d+r_0} \sum_{j=0}^{n-1} a_{0j} x_j, \ldots, \lambda^{d+r_{m-1}} \sum_{j=0}^{n-1} a_{m-1,j} x_j\right).$$

Equating the two expressions for each coordinate i gives

$$\sum_{j=0}^{n-1} a_{ij} \lambda^{q_j} x_j = \lambda^{d+r_i} \sum_{j=0}^{n-1} a_{ij} x_j.$$

Since this must hold for all $\mathbf{x}$ and all $\lambda \in k^\times$, each monomial λ^{q_j} on the left must match λ^{d+r_i} on the right wherever $a_{ij} \neq 0$. Thus

$$\lambda^{q_j} = \lambda^{d+r_i} \quad \Rightarrow \quad q_j = d + r_i \quad \Leftrightarrow \quad r_i = q_j - d.$$

Rewriting this yields $r_i = q_j + d$ as claimed.

Conversely, if this condition holds, the same calculation in reverse shows that L satisfies the homogeneity identity. □

Example 3. *For $\mathcal{V}_{(2,3)} = V_2 \oplus V_3$, a linear map $L : \mathcal{V}_{(2,3)} \to \mathcal{V}_{(2,3)}$ satisfies:*

$$L([\lambda \star \mathbf{u}]) = L([\lambda^2 f, \lambda^3 g]) = [\lambda^2 L(f), \lambda^3 L(g)] = \lambda \star [L(f), L(g)] = \lambda \star L(\mathbf{u}),$$
$$\begin{aligned} L([f,g] \oplus [f',g']) &= L([f+f', g+g']) = [L(f) + L(f'), L(g) + L(g')] \\ &= [L(f), L(g)] \oplus [L(f'), L(g')] = L([f,g]) \oplus L([f',g']). \end{aligned}$$

Using the basis $\mathcal{B} = \{x^2, xy, y^2, x^3, x^2y, xy^2, y^3\}$, where $\mathcal{B}_1 = \{x^2, xy, y^2\}$ spans V_2 and $\mathcal{B}_2 = \{x^3, x^2y, xy^2, y^3\}$ spans V_3, the polynomial

$$F(x,y) = (x^2 + xy + y^2) + (x^3 + x^2y + xy^2 + y^3)$$

has coordinates $\mathbf{u} = [1,1,1,1,1,1,1]^t$.

Scalar multiplication $L(\mathbf{x}) = \lambda \mathbf{x}$ is a graded linear map, with matrix:

$$\begin{bmatrix} \lambda^{q_0} & 0 & \cdots & 0 \\ 0 & \lambda^{q_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \lambda^{q_n} \end{bmatrix}.$$

Proposition 2. *Let $\mathcal{V}^n_{\mathbf{q}}(k)$ be a graded vector space with grading vector $\mathbf{q} = (q_0, \ldots, q_{n-1}) \in \mathbb{Q}^n_{>0}$, and scalar action*

$$\lambda \star \mathbf{x} = (\lambda^{q_0} x_0, \ldots, \lambda^{q_{n-1}} x_{n-1}).$$

Let $W \subseteq \mathcal{V}^n_{\mathbf{q}}(k)$ be a k-linear subspace. Then the following are equivalent:

- *(i) W is invariant under scalar action: $\lambda \star \mathbf{x} \in W$ for all $\lambda \in k^\times$ and all $\mathbf{x} \in W$.*
- *(ii) W is generated by homogeneous vectors in $\mathcal{V}^n_{\mathbf{q}}(k)$.*

Proof. (ii) ⇒ (i): Suppose W is spanned by homogeneous vectors $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(r)}\}$, where each $\mathbf{x}^{(j)}$ has support only on coordinates of a fixed grade. For any $\lambda \in k^\times$ and $\mathbf{x}^{(j)}$, we have:

$$\lambda \star \mathbf{x}^{(j)} = (\lambda^{q_0} x_0^{(j)}, \ldots, \lambda^{q_{n-1}} x_{n-1}^{(j)}) \in W,$$

since scalar multiplication respects homogeneity and W is a vector space. Hence, W is invariant under scalar action.

(i) ⇒ (ii): Suppose W is invariant under scalar action. Let $\mathbf{x} \in W$ be arbitrary, and write $\mathbf{x} = (x_0, \ldots, x_{n-1})$ in coordinates. For each distinct grade q appearing in $\mathbf{q}$, define a projection $\pi_q : \mathcal{V}^n_{\mathbf{q}} \to \mathcal{V}^n_{\mathbf{q}}$ by:

$$\pi_q(\mathbf{x}) = \sum_{i : q_i = q} x_i e_i,$$

where e_i is the standard basis vector in position i. Then $\mathbf{x} = \sum_q \pi_q(\mathbf{x})$, and each $\pi_q(\mathbf{x})$ is supported only on coordinates of grade q—i.e., each is homogeneous.

We claim that each $\pi_q(\mathbf{x}) \in W$. Since W is invariant, $\lambda \star \mathbf{x} \in W$ for all $\lambda \in k^\times$. Write:

$$\lambda \star \mathbf{x} = \sum_{i=0}^{n-1} \lambda^{q_i} x_i e_i = \sum_{q \in \{q_0,\ldots,q_{n-1}\}} \lambda^q \pi_q(\mathbf{x}),$$

where the sum is over distinct grades $q_1, q_2, \ldots, q_m$ in $\mathbf{q}$. To isolate $\pi_q(\mathbf{x})$, choose m distinct $\lambda_1, \ldots, \lambda_m \in k^\times$ and solve the linear system:

$$\sum_{j=1}^{m} c_j \lambda_j^{q_k} = \begin{cases} 1 & \text{if } q_k = q, \\ 0 & \text{otherwise,} \end{cases}$$

for coefficients c_j. This is a Vandermonde system, invertible since the grades q_k are distinct and $\lambda_j \neq 0$. Thus:

$$\pi_q(\mathbf{x}) = \sum_{j=1}^{m} c_j (\lambda_j \star \mathbf{x}).$$

Since each $\lambda_j \star \mathbf{x} \in W$ and W is a k-linear subspace, $\pi_q(\mathbf{x}) \in W$. Hence, $\mathbf{x} = \sum_q \pi_q(\mathbf{x})$ is a sum of homogeneous vectors in W, and W is generated by homogeneous vectors. □

Corollary 1. *Let $V = \bigoplus_{d \in I} V_d$ be an I-graded vector space over a field k, where each V_d consists of homogeneous elements of degree d. Then for each $d \in I$, the subspace V_d is a maximal subspace of V invariant under scalar action*

$$\lambda \star v = \lambda^d v.$$

Moreover, any proper grading-invariant subspace $W \subset V$ contained in V_d is necessarily a k-subspace of V_d and hence not grading-invariant unless $W = V_d$.

Proof. By definition of grading, any $v \in V_d$ satisfies $\lambda \star v = \lambda^d v$, so V_d is invariant under scalar action.

Suppose $W \subseteq V$ is any subspace invariant under the scalar action and contained in V_d. Then by Prop. 2, W must be generated by homogeneous vectors, and since the only homogeneous vectors in V_d have degree d, we conclude $W \subseteq V_d$.

To show maximality: suppose $W \supsetneq V_d$ and is invariant. Then W must contain some element v with a nonzero component outside of V_d. But then its homogeneous decomposition contains terms of other degrees, contradicting that W is contained in V_d. Hence, V_d is the largest subspace consisting entirely of degree-d homogeneous elements and invariant under the action. □

2.3. **Operations over Graded Vector Spaces.** For I-graded spaces $V = \bigoplus_{i \in I} V_i$ and $W = \bigoplus_{i \in I} W_i$, the **direct sum** is $V \oplus W$ with gradation:

$$(V \oplus W)_i = V_i \oplus W_i.$$

Scalar multiplication is $\lambda(v_i, w_i) = (\lambda v_i, \lambda w_i)$. For differing grade sets I and J, index over $I \cup J$, with $(V \oplus W)_k = V_k \oplus W_k$ ($V_k = 0$ if $k \notin I$).

Consider two graded vector spaces $V = \bigoplus_{i \in I} V_i$ and $W = \bigoplus_{i \in I} W_i$, where I is a semigroup (e.g., $\mathbb{N}$ with addition). The **tensor product** $V \otimes W$ is a graded vector space with components:

$$(V \otimes W)_i = \bigoplus_{(j,k): j+k=i} (V_j \otimes W_k),$$

where $v_j \in V_j$ and $w_k \in W_k$ form $v_j \otimes w_k$ of grade $j+k$, and scalar multiplication is given by $\lambda(v_j \otimes w_k) = (\lambda v_j) \otimes w_k$.

For non-semigroup I (e.g., $\mathbb{Q}$), the tensor product adapts by defining grades via a suitable operation, ensuring grading consistency; see [1].

Example 4. *For example, take $V = \mathcal{V}_{(2,3)} = V_2 \oplus V_3$, with V_2 and V_3 as spaces of quadratic and cubic polynomials, respectively. The tensor product $V \otimes V$ is:*

$$\mathcal{V}_{(2,3)} \otimes \mathcal{V}_{(2,3)} = (V_2 \otimes V_2)_4 \oplus (V_2 \otimes V_3)_5 \oplus (V_3 \otimes V_2)_5 \oplus (V_3 \otimes V_3)_6.$$

If $f = x^2 \in V_2$ (grade 2) and $g = x^3 \in V_3$ (grade 3), then $f \otimes g \in (V_2 \otimes V_3)_5$, since $2+3=5$.

For $\mathcal{V}_{(1/2,1/3)}$, the tensor product yields grades like $1/2 + 1/3 = 5/6$, illustrating fractional gradations.

For the coordinate-wise space $\mathcal{V}^n_{\mathbf{q}}(k) = k^n$ with $\mathrm{gr}(x_i) = q_i$, the tensor product with $\mathcal{V}^m_{\mathbf{q}'}(k) = k^m$ (grades $\mathrm{gr}(x'_j) = q'_j$) is:

$$\mathcal{V}^n_{\mathbf{q}} \otimes \mathcal{V}^m_{\mathbf{q}'} = \bigoplus_{i=0}^{n-1} \bigoplus_{j=0}^{m-1} k(e_i \otimes e'_j),$$

where $e_i \otimes e'_j$ has grade $q_i + q'_j$. This form accommodates varying grades across spaces, relevant to inputs of differing significance.

For three graded vector spaces U, V, and W over a semigroup I, the tensor product is **associative**: $(U \otimes V) \otimes W \cong U \otimes (V \otimes W)$. The graded components of $(U \otimes V) \otimes W$ are:

$$((U \otimes V) \otimes W)_i = \bigoplus_{(j,k,l):j+k+l=i} (U_j \otimes V_k) \otimes W_l,$$

where j, k, and l are grades in U, V, and W, respectively. This property ensures consistency in composing multiple tensor operations, analogous to stacking transformations in neural network layers.

Proposition 3. *Let $\mathcal{V}^n_{\mathbf{q}}(k)$ and $\mathcal{V}^m_{\mathbf{q}'}(k)$ be coordinate-wise graded vector spaces with grading vectors $\mathbf{q} = (q_0, \ldots, q_{n-1})$ and $\mathbf{q}' = (r_0, \ldots, r_{m-1})$, respectively, where $q_i, r_j \in \mathbb{Q}_{>0}$. Then the tensor product*

$$\mathcal{V}^n_{\mathbf{q}}(k) \otimes \mathcal{V}^m_{\mathbf{q}'}(k)$$

inherits a natural $\mathbb{Q}$-grading with basis elements $e_i \otimes e'_j$ having grade $q_i + r_j$. That is,

$$\mathrm{gr}(e_i \otimes e'_j) = q_i + r_j,$$

and the resulting graded vector space is

$$\mathcal{V}^{nm}_{\mathbf{q} \boxplus \mathbf{q}'}(k), \quad \text{where } \mathbf{q} \boxplus \mathbf{q}' = \{q_i + r_j \mid 0 \le i < n,\ 0 \le j < m\}.$$

Proof. Let $\{e_i\}_{i=0}^{n-1}$ and $\{e'_j\}_{j=0}^{m-1}$ be the standard bases of $\mathcal{V}^n_{\mathbf{q}}(k)$ and $\mathcal{V}^m_{\mathbf{q}'}(k)$, respectively, with $\mathrm{gr}(e_i) = q_i$ and $\mathrm{gr}(e'_j) = r_j$.

By bilinearity of the tensor product, the elements $\{e_i \otimes e'_j\}$ form a basis for $\mathcal{V}^n_{\mathbf{q}} \otimes \mathcal{V}^m_{\mathbf{q}'}$. Define the grading on the tensor product by declaring

$$\mathrm{gr}(e_i \otimes e'_j) = \mathrm{gr}(e_i) + \mathrm{gr}(e'_j) = q_i + r_j.$$

This grading is additive and extends linearly: any tensor $v = \sum_{i,j} a_{ij} e_i \otimes e'_j$ is a sum of homogeneous elements with well-defined grades.

We now check compatibility with scalar action. Let $\lambda \in k^\times$. On $\mathcal{V}_{\mathbf{q}}^n$, we have

$$\lambda \star e_i = \lambda^{q_i} e_i, \quad \text{and similarly on } \mathcal{V}_{\mathbf{q}'}^m : \quad \lambda \star e'_j = \lambda^{r_j} e'_j.$$

The induced action on the tensor product satisfies:

$$\lambda \star (e_i \otimes e'_j) = (\lambda \star e_i) \otimes (\lambda \star e'_j) = \lambda^{q_i} \lambda^{r_j} (e_i \otimes e'_j) = \lambda^{q_i + r_j} (e_i \otimes e'_j).$$

Hence, the scalar action respects the grading defined above, and $\mathcal{V}_{\mathbf{q}}^n \otimes \mathcal{V}_{\mathbf{q}'}^m$ becomes a graded vector space.

The multiset of grades $\mathbf{q} \boxplus \mathbf{q}'$ records the full set of values $q_i + r_j$, indexed by pairs (i, j), forming a grading vector of length nm for the tensor space. □

Example 5. *Let $\mathcal{V}_{\mathbf{q}}^2(k)$ and $\mathcal{V}_{\mathbf{q}'}^2(k)$ be graded vector spaces with $\mathbf{q} = (1, 2)$ and $\mathbf{q}' = (3, 4)$. The basis of $\mathcal{V}_{\mathbf{q}}^2$ is $\{e_0, e_1\}$ with $\text{gr}(e_0) = 1$ and $\text{gr}(e_1) = 2$, and similarly for $\mathcal{V}_{\mathbf{q}'}^2$ with basis $\{e'_0, e'_1\}$ and grades 3, 4.*

The tensor product $\mathcal{V}_{\mathbf{q}}^2 \otimes \mathcal{V}_{\mathbf{q}'}^2$ has basis:

$$\{e_0 \otimes e'_0,\ e_0 \otimes e'_1,\ e_1 \otimes e'_0,\ e_1 \otimes e'_1\}.$$

Each basis element is homogeneous, with grades computed by summing the component grades:

$$\begin{aligned} \text{gr}(e_0 \otimes e'_0) &= 1 + 3 = 4, \\ \text{gr}(e_0 \otimes e'_1) &= 1 + 4 = 5, \\ \text{gr}(e_1 \otimes e'_0) &= 2 + 3 = 5, \\ \text{gr}(e_1 \otimes e'_1) &= 2 + 4 = 6. \end{aligned}$$

Thus, the tensor product decomposes as:

$$\mathcal{V}_{\mathbf{q}}^2 \otimes \mathcal{V}_{\mathbf{q}'}^2 = V_4 \oplus V_5 \oplus V_6,$$

where:

$$\begin{aligned} V_4 &= \mathit{span}_k\{e_0 \otimes e'_0\}, \\ V_5 &= \mathit{span}_k\{e_0 \otimes e'_1,\ e_1 \otimes e'_0\}, \\ V_6 &= \mathit{span}_k\{e_1 \otimes e'_1\}. \end{aligned}$$

This example shows that the set of grades is $\{4, 5, 6\}$, and the multiplicity of each grade reflects how many pairs (i, j) satisfy $q_i + r_j = d$. Note in particular that grade 5 appears twice, from two distinct tensor combinations. The tensor product is naturally graded but not necessarily multiplicity-free. In the context of graded neural networks, this multiplicity implies that layers processing inputs in V_5 must account for a two-dimensional subspace, potentially requiring distinct neurons or weight adjustments to differentiate contributions from $e_0 \otimes e'_1$ and $e_1 \otimes e'_0$. For example, a graded neuron $\alpha_q(\mathbf{x}) = \sum w_i^{q_i} x_i + b$ may assign separate weights to each basis element of grade 5 to preserve the graded structure, impacting computational complexity and expressivity.

Next, consider the **dual space** of $V = \bigoplus_{i \in I} V_i$, where I is a general index set (e.g., $\mathbb{N}$, $\mathbb{Z}$), not necessarily a semigroup. The **dual** $V^* = \text{Hom}_k(V, k)$ is graded as:

$$V^* = \bigoplus_{i \in I} V_{-i}^*,$$

with $V_{-i}^* = \{f : V \to k \mid f(V_i) \subseteq k, f(V_j) = 0 \text{ if } j \neq i\}$. The grade $-i$ arises because a functional on V_i (grade i) pairs to produce a scalar (grade 0), requiring $i + (-i) = 0$.

For $I = \mathbb{Q}$, dual grades $-i$ ensure scalar compatibility, critical for defining graded loss functions; see [1].

For $\mathcal{V}_{\mathbf{q}}^n(k) = k^n$ with $\mathrm{gr}(x_i) = q_i$ and scalar action $\lambda \star \mathbf{x} = (\lambda^{q_i} x_i)$, the dual $(\mathcal{V}_{\mathbf{q}}^n)^* = k^n$ has basis functionals f_i of grade $\mathrm{gr}(f_i) = -q_i$, with $\lambda \star f_i = \lambda^{-q_i} f_i$. This inverse scaling complements the original action, suggesting applications in defining graded loss functions or optimization procedures for neural networks.

2.4. **Inner Graded Vector Spaces and their Norms.** Consider now the case when each V_i is a finite-dimensional inner space, and let $\langle \cdot, \cdot \rangle_i$ denote the corresponding inner product. Then we can define an inner product on $V = \bigoplus_{i \in I} V_i$ as follows. For $\mathbf{u} = u_1 + \ldots + u_n$ and $\mathbf{v} = v_1 + \ldots + v_n$, where $u_i, v_i \in V_i$, we define:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \langle u_1, v_1 \rangle_1 + \ldots + \langle u_n, v_n \rangle_n,$$

which is the standard product across graded components. The Euclidean norm is then:

$$\|\mathbf{u}\| = \sqrt{u_1^2 + \ldots + u_n^2},$$

where $\|u_i\|_i = \sqrt{\langle u_i, u_i \rangle_i}$ is the norm in V_i, and we assume an orthonormal basis for simplicity. For non-integer I (e.g., $\mathbb{Q}$), norms may incorporate grade weights, e.g., $\|\mathbf{x}\|_{\mathbf{q}} = \left(\sum i |x_i|^2 \right)^{1/2}$ for $i \in \mathbb{Q}$.

If such V_i are not necessarily finite-dimensional, then we have to assume that V_i is a Hilbert space (i.e., a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product).

Example 6. *Let us continue with the space $\mathcal{V}_{(2,3)}$ with bases $\mathcal{B}_1 = \{x^2, xy, y^2\}$ for V_2 and $\mathcal{B}_2 = \{x^3, x^2y, xy^2, y^3\}$ for V_3, as in Example 3. Hence, a basis for $\mathcal{V}_{(2,3)} = V_2 \oplus V_3$ is $\mathcal{B} = \{x^2, xy, y^2, x^3, x^2y, xy^2, y^3\}$. Let $\mathbf{u}, \mathbf{v} \in \mathcal{V}_{(2,3)}$ be given by:*

$$\mathbf{u} = \mathbf{a} + \mathbf{b} = \left(u_1 x^2 + u_2 xy + u_3 y^2\right) + \left(u_4 x^3 + u_5 x^2 y + u_6 xy^2 + u_7 y^3\right),$$
$$\mathbf{v} = \mathbf{a}' + \mathbf{b}' = \left(v_1 x^2 + v_2 xy + v_3 y^2\right) + \left(v_4 x^3 + v_5 x^2 y + v_6 xy^2 + v_7 y^3\right).$$

Then

$$\begin{aligned} \langle \mathbf{u}, \mathbf{v} \rangle &= \langle \mathbf{a} + \mathbf{b}, \mathbf{a}' + \mathbf{b}' \rangle = \langle \mathbf{a}, \mathbf{a}' \rangle_2 + \langle \mathbf{b}, \mathbf{b}' \rangle_3 \\ &= u_1 v_1 + u_2 v_2 + u_3 v_3 + u_4 v_4 + u_5 v_5 + u_6 v_6 + u_7 v_7, \end{aligned}$$

and the Euclidean norm is

$$\|\mathbf{u}\| = \sqrt{u_1^2 + \ldots + u_7^2},$$

assuming $\mathcal{B}$ is orthonormal.

For $\mathcal{V}_{\left(\frac{1}{2}, \frac{1}{3}\right)}$, a weighted norm like

$$\|\mathbf{u}\|_{\mathbf{q}} = \sqrt{\frac{1}{2} u_1^2 + \frac{1}{3} u_2^2}$$

could prioritize fractional grades.

There are other ways to define a norm on graded spaces, particularly to emphasize the grading. Consider a Lie algebra $\mathfrak{g}$ called **graded** if there is a finite family of subspaces $V_1, \ldots, V_r$ such that $\mathfrak{g} = V_1 \oplus \cdots \oplus V_r$ and $[V_i, V_j] \subset V_{i+j}$, where $[V_i, V_j]$ is the Lie bracket. When $\mathfrak{g}$ is graded, define a dilation for $t \in \mathbb{R}^\times$, $\alpha_t : \mathfrak{g} \to \mathfrak{g}$, by:

$$\alpha_t(v_1, \ldots, v_r) = (tv_1, t^2 v_2, \ldots, t^r v_r).$$

We define a **homogeneous norm** on $\mathfrak{g}$ as

$$\|\mathbf{v}\| = \|(v_1, \ldots, v_r)\| = \left(\|v_1\|_1^{2r} + \|v_2\|_2^{2r-2} + \cdots + \|v_r\|_r^2\right)^{1/2r}, \tag{1}$$

where $\|\cdot\|_i$ is the Euclidean norm on V_i, and $r = \max\{i\}$. This norm is homogeneous under α_t: $\|\alpha_t(\mathbf{v})\| = |t|\|\mathbf{v}\|$, reflecting the grading grades. It satisfies the triangle inequality, as shown in [6], and is detailed in [7, 8]. For $\mathcal{V}_{(2,3)}$ with $r = 3$, if $\mathbf{u} = (u_1, u_2) \in V_2 \oplus V_3$, then:

$$\|\mathbf{u}\| = \left(\|u_1\|_2^6 + \|u_2\|_3^2\right)^{1/6},$$

giving higher weight to lower-degree components. A more general approach is considered in [9], defining norms for line bundles and weighted heights on weighted projective varieties.

Definition. 1. *For $\mathcal{V}_{\mathbf{q}}^n(k) = k^n$ with $\operatorname{gr}(x_i) = q_i$, a **graded Euclidean norm** can be:*

$$\|\mathbf{x}\|_{\mathbf{q}} = \left(\sum_{i=0}^{n-1} q_i |x_i|^2\right)^{1/2}, \tag{2}$$

weighting each coordinate by its grade q_i.

Alternatively, a **max-graded norm** is:

$$\|\mathbf{x}\|_{\max} = \max_i \{q_i^{1/2} |x_i|\}, \tag{3}$$

emphasizing the dominant graded component, akin to L_∞ norms but adjusted by q_i.

Example 7. *For $\mathbf{x} = (x_1, x_2) \in \mathcal{V}_{(2,3)}$ with coordinates in basis $\mathcal{B}$, let $x_1 = (1, 0, 1) \in V_2$, $x_2 = (1, -1, 0, 1) \in V_3$. The graded Euclidean norm is:*

$$\|\mathbf{x}\|_{\mathbf{q}} = \left(2(1^2 + 0^2 + 1^2) + 3(1^2 + (-1)^2 + 0^2 + 1^2)\right)^{1/2} = \sqrt{2 \cdot 2 + 3 \cdot 3} = \sqrt{13},$$

while the max-graded norm is:

$$\|\mathbf{x}\|_{max} = \max\{2^{1/2} \cdot 1, 2^{1/2} \cdot 0, 2^{1/2} \cdot 1, 3^{1/2} \cdot 1, 3^{1/2} \cdot 1, 3^{1/2} \cdot 0, 3^{1/2} \cdot 1\} = 3^{1/2}.$$

These differ from the standard $\|\mathbf{x}\| = \sqrt{6}$, highlighting grading's impact.

Remark 1 (Properties of Graded Norms.)**.** *The graded Euclidean norm $\|\cdot\|_{\mathbf{q}}$ is a true norm:*

(i) $\|\mathbf{x}\|_{\mathbf{q}} \geq 0$, zero iff $\mathbf{x} = 0$;
(ii) $\|\lambda\mathbf{x}\|_{\mathbf{q}} = |\lambda|\|\mathbf{x}\|_{\mathbf{q}}$;
(iii) $\|\mathbf{x} + \mathbf{y}\|_{\mathbf{q}} \leq \|\mathbf{x}\|_{\mathbf{q}} + \|\mathbf{y}\|_{\mathbf{q}}$ (via Cauchy-Schwarz).

The homogeneous norm $\|\cdot\|$ is also a norm, satisfying similar properties under the dilation α_t, and is differentiable except at zero; see [6]. The max-graded norm satisfies norm axioms but is less smooth. These norms extend to infinite I in Hilbert spaces with convergence conditions; see [7].

Definition. 2. *A norm $\|\cdot\|$ is **convex** if for all $\mathbf{x}, \mathbf{y} \in V$ and $t \in [0, 1]$,*

$$\|t\mathbf{x} + (1-t)\mathbf{y}\| \leq t\|\mathbf{x}\| + (1-t)\|\mathbf{y}\|.$$

The Euclidean norm $\|\mathbf{x}\| = \sqrt{\sum x_i^2}$ is convex, as its square $\|\mathbf{x}\|^2$ is quadratic with Hessian $\nabla^2(\|\mathbf{x}\|^2) = 2I$, positive definite.

For the graded Euclidean norm

$$\|\mathbf{x}\|_{\mathbf{q}} = \left(\sum q_i |x_i|^2\right)^{1/2}$$

with $q_i > 0$, let $f(\mathbf{x}) = \|\mathbf{x}\|_{\mathbf{q}}^2 = \sum q_i |x_i|^2$; the Hessian is $\nabla^2 f = 2\text{diag}(q_0, \ldots, q_{n-1})$, positive definite, so $\|\cdot\|_{\mathbf{q}}$ is convex.

The homogeneous norm

$$\|\mathbf{v}\| = \left(\sum \|v_i\|_i^{2r-2(i-1)}\right)^{1/2r}$$

is less straightforward. For example, for $\mathcal{V}_{(2,3)}$ ($r = 3$), $\|\mathbf{u}\| = (\|u_1\|_2^6 + \|u_2\|_3^2)^{1/6}$. Define

$$f(\mathbf{u}) = \|\mathbf{u}\|^6 = \|u_1\|_2^6 + \|u_2\|_3^2;$$

the Hessian includes $\partial^2 f/\partial u_{1j}^2 = 30u_{1j}^4$, positive for $\mathbf{u} \neq 0$, but near zero, high exponents (i.e., 6) disrupt convexity. However, $\|\mathbf{u}\|$ is quasiconvex, as sublevel sets $\{\mathbf{u} \mid \|\mathbf{u}\| \leq c\}$ are convex for $c > 0$ (see [6]), reflecting a weaker but useful property.

The max-graded norm

$$\|\mathbf{x}\|_{\max} = \max\{q_i^{1/2}|x_i|\}$$

is convex, as the maximum of convex functions $q_i^{1/2}|x_i|$, with sublevel sets being intersections of slabs $\{\mathbf{x} \mid q_i^{1/2}|x_i| \leq c\}$; see [10].

Gradient behavior is analyzed via the function $f(\mathbf{x}) = \|\mathbf{x}\|^2$. For the Euclidean norm, $f(\mathbf{x}) = \sum x_i^2$, $\nabla f = 2\mathbf{x}$, linear and isotropic. For $\|\cdot\|_{\mathbf{q}}$,

$$f(\mathbf{x}) = \sum q_i |x_i|^2,$$

$\nabla f = 2(q_0 x_0, \ldots, q_{n-1} x_{n-1})$, scaling components by q_i, with magnitude

$$\|\nabla f\|_2 = 2\sqrt{\sum q_i^2 x_i^2}.$$

For the homogeneous norm on $\mathcal{V}_{(2,3)}$,

$$f(\mathbf{u}) = \|u_1\|_2^6 + \|u_2\|_3^2,$$

where $\nabla f = (6\|u_1\|_2^4 u_1, 2u_2)$, nonlinear with steep growth in V_2 (exponent 4) versus V_3 (exponent 1).

The max-graded norm's

$$f(\mathbf{x}) = (\max q_i^{1/2}|x_i|)^2$$

has a subdifferential, i.e.,

$$\partial f/\partial x_i = 2q_i^{1/2}\text{sgn}(x_i)\max\{q_j^{1/2}|x_j|\}$$

if i achieves the max, zero otherwise, reflecting discontinuity; see [10].

2.5. **Graded and Filtered Structures.** The algebraic notion of grading is closely related to filtrations. In fact, under suitable conditions, graded and filtered vector spaces can be viewed as two sides of the same structure.

Lemma 1. *Let V be a k-vector space.*

(i) Every increasing filtration

$$0 = F^{-1}V \subseteq F^0V \subseteq F^1V \subseteq \cdots \subseteq V,$$

that is exhaustive ($\bigcup_i F^iV = V$) and separated ($\bigcap_i F^iV = 0$), induces a graded vector space

$$\mathrm{gr}(V) = \bigoplus_i \mathrm{gr}_i(V), \quad \mathrm{gr}_i(V) := F^iV/F^{i-1}V.$$

(ii) Conversely, any $\mathbb{Z}$-graded vector space $V = \bigoplus_{i\in\mathbb{Z}} V_i$ admits a canonical increasing filtration

$$F^nV := \bigoplus_{i\leq n} V_i,$$

whose associated graded space is isomorphic to V.

This correspondence allows one to move between additive decompositions and nested hierarchical representations. In many applications, such as optimization, PDEs, and signal processing, filtered structures naturally encode progressive refinement. In the context of neural networks, especially Graded Neural Networks, this algebraic link suggests a deep geometric and architectural interpretation.

2.6. **Learning from Coarse to Fine.** In the GNN framework, feature coordinates are graded: components with low grades (e.g., $q_i = 1, 2$) correspond to coarse, high-level representations—global symmetries, low-degree features, or dominant structure—while higher-grade components ($q_i \gg 1$) encode fine-grained, localized, or higher-frequency detail.

This mirrors the classical multiresolution paradigm, where models learn progressively refined representations. A filtration $F^0 \subset F^1 \subset \ldots$ naturally encodes such depth or semantic scale, and its associated graded structure allows explicit control over what level of detail a layer or operation is sensitive to.

For instance, in applications to symbolic algebra (e.g., computing invariants of curves), lower-graded components dominate global structure (e.g., J_2 and J_4), while higher-graded ones reflect subtle moduli (e.g., J_{10}). GNNs trained on such data are implicitly performing filtered learning—starting with robust, coarse predictors and gradually refining toward higher-grade features.

This perspective aligns naturally with curriculum learning, progressive training, or hierarchical inference, and could inform both architecture design (e.g., grade-specific layers) and optimization strategies (e.g., prioritizing coarse loss components early in training).

3. Graded Neural Networks (GNN)

We define artificial neural networks over graded vector spaces, utilizing Section 2. Let k be a field, and for $n \geq 1$, denote $\mathbb{A}^n_k$ (resp. $\mathbb{P}^n_k$) as the affine (resp. projective) space over k, omitting the subscript if k is algebraically closed. A tuple $\mathbf{q} =$

$(q_0, \ldots, q_{n-1}) \in \mathbb{N}^n$ defines the **grades**, with $\text{gr}(x_i) = q_i$. The graded vector space $\mathcal{V}_{\mathbf{q}}^n(k) = k^n$ has scalar multiplication:

$$\lambda \star \mathbf{x} = (\lambda^{q_0} x_0, \ldots, \lambda^{q_{n-1}} x_{n-1}), \quad \mathbf{x} = (x_0, \ldots, x_{n-1}) \in k^n, \lambda \in k,$$

as in Section 2, denoted $\mathcal{V}_{\mathbf{q}}$ when clear. This scalar action, denoted $\lambda \star \mathbf{x}$, mirrors the graded multiplication in Section 2, applicable to both the coordinate form here and the direct sum form (e.g., $\lambda \star [f, g]$) via basis representation.

A **graded neuron** on $\mathcal{V}_{\mathbf{q}}$ is typically defined as an additive map $\alpha_{\mathbf{q}} : \mathcal{V}_{\mathbf{q}}^n \to k$ such that

$$\alpha_{\mathbf{q}}(\mathbf{x}) = \sum_{i=0}^{n-1} w_i^{q_i} x_i + b,$$

where $w_i \in k$ are **neural weights**, and $b \in k$ is the **bias**. For $b = 0$,

$$\alpha_{\mathbf{q}}(\lambda \star \mathbf{x}) = \sum (\lambda w_i)^{q_i} x_i = \lambda \sum w_i' x_i$$

for $(w_i' = w_i^{q_i})$, approximating a graded linear map of degree 1 per Section 2. With $b \neq 0$, $\alpha_{\mathbf{q}}$ is affine, embedding grading via $w_i^{q_i}$. Alternatively, a **multiplicative graded neuron** can be defined as $\beta_{\mathbf{q}} : \mathcal{V}_{\mathbf{q}}^n \to k$ such that

$$\beta_{\mathbf{q}}(\mathbf{x}) = \prod_{i=0}^{n-1} (w_i x_i)^{q_i} + b,$$

capturing multiplicative interactions among graded features, suitable for tasks like polynomial modeling in [1]. For $b = 0$,

$$\beta_{\mathbf{q}}(\lambda \star \mathbf{x}) = \prod (\lambda^{q_i} w_i x_i)^{q_i} = \lambda^{\sum q_i^2} \prod (w_i x_i)^{q_i},$$

reflecting a higher-degree graded map, enhancing expressivity for nonlinear relationships.

A **graded network layer** is:

$$\begin{aligned} \phi : \mathcal{V}_{\mathbf{q}}^n(k) &\to \mathcal{V}_{\mathbf{q}}^n(k) \\ \mathbf{x} &\to g(W\mathbf{x} + \mathbf{b}), \end{aligned}$$

where $W = [w_{j,i}^{q_i}] \in k^{n \times n}$, $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in k^n$, and ϕ preserves grading, with $\text{gr}(y_j) = q_j$. Layers using multiplicative neurons, $\phi(\mathbf{x}) = g(\prod (W\mathbf{x})^{q_i} + \mathbf{b})$, are also possible but increase computational complexity; see [1].

Remark 2. *Neural weights w_i or $w_{j,i}$ differ from grades q_i. Exponents $w_i^{q_i}$ (or $(w_i x_i)^{q_i}$ in multiplicative neurons) reflect grading, while q_i define $\mathcal{V}_{\mathbf{q}}$'s action. We use w for weights, q_i for grades.*

A **graded neural network** (GNN) is a composition of multiple layers given as

$$\hat{\mathbf{y}} = \phi_m \circ \cdots \circ \phi_1(\mathbf{x}),$$

where each layer $\phi_l(\mathbf{x}) = g_l(W^l \mathbf{x} + \mathbf{b}^l)$ applies a transformation defined by the matrix of neural weights $W^l = [w_{j,i}^{q_i}]$, producing outputs $\hat{\mathbf{y}}$ and true values $\mathbf{y}$ in $\mathcal{V}_{\mathbf{q}}^n$ with grades $\text{gr}(\hat{y}_i) = q_i$. Hybrid GNNs combining additive and multiplicative neurons across layers are also viable, offering flexibility for diverse applications.

3.1. **ReLU Activation.** In classical neural networks, the rectified linear unit (ReLU) activation, defined as $\text{ReLu}(x) = \max\{0, x\}$, applies a simple thresholding to promote sparsity and efficiency. However, for graded neural networks over $\mathcal{V}_{\mathbf{q}}^n$, where $\mathbf{x} = (x_0, \ldots, x_{n-1})$ has coordinates with grades $\text{gr}(x_i) = q_i$ and scalar action $\lambda \star \mathbf{x} = (\lambda^{q_0} x_0, \ldots, \lambda^{q_{n-1}} x_{n-1})$, a direct application of this ReLU ignores the grading's intrinsic scaling. To adapt to this structure, we define a *graded ReLU* that adjusts nonlinearity by grade. For $\mathbf{x} \in \mathcal{V}_{\mathbf{q}}^n$, the graded ReLU is:

$$\text{ReLu}_i(x_i) = \max\{0, |x_i|^{1/q_i}\},$$

and

$$\text{ReLu}(\mathbf{x}) = (\text{ReLu}_0(x_0), \ldots, \text{ReLu}_{n-1}(x_{n-1})).$$

Unlike the classical $\max\{0, x_i\}$, which treats all coordinates uniformly, this version scales each x_i by $1/q_i$, reflecting the graded action. For $\lambda \star \mathbf{x} = (\lambda^{q_i} x_i)$, compute:

$$\text{ReLu}_i(\lambda^{q_i} x_i) = \max\{0, |\lambda^{q_i} x_i|^{1/q_i}\} = \max\{0, |\lambda||x_i|^{1/q_i}\} = |\lambda| \max\{0, |x_i|^{1/q_i}\},$$

so $\text{ReLu}(\lambda \star \mathbf{x}) = |\lambda| \, \text{ReLu}(\mathbf{x})$ for $\lambda > 0$, aligning with $\mathcal{V}_{\mathbf{q}}^n$'s grading up to magnitude. This ensures the activation respects the differential scaling of coordinates (i.e., $q_i = 2$ vs. $q_i = 3$ in $\mathcal{V}_{(2,3)}$), unlike the classical ReLU, where $\text{ReLu}(\lambda x_i) = \lambda \, \text{ReLu}(x_i)$ for $\lambda > 0$ assumes homogeneity of degree 1.

Remark 3. *The graded ReLU outputs positive values for negative inputs (e.g.,* $\text{ReLu}_i(-x_i) = |x_i|^{1/q_i}$*), unlike the classical ReLU, which outputs 0. This emphasizes magnitude scaling, preserving the graded structure, but may reduce sparsity, potentially affecting training efficiency in GNNs. Alternative definitions, such as* $\max\{0, |x_i|^{1/q_i} \operatorname{sgn}(x_i)\}$*, could restore thresholding for negative inputs while maintaining graded scaling.*

An alternative **exponential graded activation** is defined as:

$$\exp_i(x_i) = \exp\left(\frac{x_i}{q_i}\right) - 1,$$

and

$$\exp(\mathbf{x}) = (\exp_0(x_0), \ldots, \exp_{n-1}(x_{n-1})).$$

This activation mitigates numerical instability for large q_i by scaling inputs inversely, ensuring smoother gradients. For $\lambda \star \mathbf{x}$,

$$\exp_i(\lambda^{q_i} x_i) = \exp\left(\frac{\lambda^{q_i} x_i}{q_i}\right) - 1,$$

which grows more gradually than ReLu_i, enhancing stability in deep GNNs.

This adaptation is motivated by the need to capture feature significance in graded spaces, as seen in applications like genus two curve invariants (J_2, J_4, J_6, J_{10} with grades 2, 4, 6, 10). A classical ReLU might underweight high-graded features (i.e., J_{10}) or overreact to low-graded ones (i.e., J_2), whereas the graded ReLU normalizes sensitivity via $1/q_i$, akin to the homogeneous norm's scaling in Section 2. The exponential activation further stabilizes high-grade features, making it suitable for tasks like quantum state modeling; see [1]. Both activations mirror weighted heights from [9, 11], where exponents adjust to graded geometry.

Example 8. *Consider $\mathcal{V}_{(2,3)}$ from Example 1, with $\mathbf{q} = (2,2,2,3,3,3,3)$ and basis*

$$\mathcal{B} = \{x^2, xy, y^2, x^3, x^2y, xy^2, y^3\}.$$

Let $\mathbf{u} = (2,-3,1,1,-2,1,1)$, representing the coordinates of a polynomial $\mathbf{u} = [f,g] \in V_2 \oplus V_3$ in the basis $\mathcal{B}$ from Example 1, mapping $f = 2x^2 - 3xy + y^2$ and $g = x^3 - 2x^2y + xy^2 + y^3$ to $\mathbb{R}^7$. Applying the graded ReLU:

$$\begin{aligned}
\mathrm{ReLu}_0(2) &= \max\{0, |2|^{1/2}\} = \sqrt{2} \approx 1.414,\\
\mathrm{ReLu}_1(-3) &= \max\{0, |-3|^{1/2}\} = \sqrt{3} \approx 1.732,\\
\mathrm{ReLu}_2(1) &= \max\{0, |1|^{1/2}\} = 1,\\
\mathrm{ReLu}_3(1) &= \max\{0, |1|^{1/3}\} = 1,\\
\mathrm{ReLu}_4(-2) &= \max\{0, |-2|^{1/3}\} = \sqrt[3]{2} \approx 1.260,\\
\mathrm{ReLu}_5(1) &= \max\{0, |1|^{1/3}\} = 1,\\
\mathrm{ReLu}_6(1) &= \max\{0, |1|^{1/3}\} = 1.
\end{aligned}$$

Thus:

$$\mathrm{ReLu}(\mathbf{u}) = (\sqrt{2}, \sqrt{3}, 1, 1, \sqrt[3]{2}, 1, 1) \approx (1.414, 1.732, 1, 1, 1.260, 1, 1).$$

For the exponential activation:

$$exp(\mathbf{u}) = (e^{2/2} - 1, e^{-3/2} - 1, e^{1/2} - 1, e^{1/3} - 1, e^{-2/3} - 1, e^{1/3} - 1, e^{1/3} - 1),$$

e.g., $exp_0(2) = e - 1 \approx 1.718$, $exp_1(-3) = e^{-1.5} - 1 \approx -0.777$, $exp_3(1) = e^{1/3} - 1 \approx 0.395$. Compare to classical ReLU: $\mathrm{ReLu}(-3) = 0$, $\mathrm{ReLu}(2) = 2$, *yielding $(2,0,1,1,0,1,1)$, which loses the graded nuance (e.g., $-3 \to \sqrt{3}$ vs. 0). The graded ReLU preserves $\mathcal{V}_{\mathbf{q}}^n$'s structure while adjusting output scale, while the exponential activation ensures smoother outputs for large q_i.*

The graded ReLU and exponential activations balance nonlinearity with grading, enhancing feature discrimination in $\mathcal{V}_{\mathbf{q}}^n$ compared to the uniform thresholding of classical ReLU. Their efficiency relative to other adaptations (e.g., $\max\{0, x_i/q_i\}$) remains to be explored, but their forms leverage the algebraic structure established in Section 2.

3.2. **Graded Loss Functions.** In classical neural networks, loss functions like the mean squared error (MSE), $L = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2$, treat all coordinates equally, assuming a uniform vector space structure. In contrast, for graded neural networks over $\mathcal{V}_{\mathbf{q}}^n(k) = k^n$ with grading $\mathrm{gr}(x_i) = q_i$ and scalar action $\lambda \star \mathbf{x} = (\lambda^{q_0}x_0, \ldots, \lambda^{q_{n-1}}x_{n-1})$, graded loss functions incorporate the grading vector $\mathbf{q} = (q_0, \ldots, q_{n-1})$ to prioritize errors based on feature significance. This enhances GNN training for hierarchical data, as evidenced by improved accuracy (e.g., 99% for genus two invariants in [2]). We define norm-based losses for regression tasks, which emphasize error magnitude, and classification losses for categorical tasks, all leveraging q_i to align with $\mathcal{V}_{\mathbf{q}}^n$'s algebraic structure.

Definition. 3. *The **graded MSE** on $\mathcal{V}_{\mathbf{q}}^n$ is:*

$$L_{MSE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=0}^{n-1} q_i(y_i - \hat{y}_i)^2,$$

where $\mathbf{y}, \hat{\mathbf{y}} \in \mathcal{V}_{\mathbf{q}}^n$ are true and predicted values, and q_i amplifies errors for higher-graded coordinates.

This scales with grading: for $\lambda \star (\mathbf{y} - \hat{\mathbf{y}}) = (\lambda^{q_i}(y_i - \hat{y}_i))$,

$$L_{\text{MSE}}(\lambda \star \mathbf{y}, \lambda \star \hat{\mathbf{y}}) = \frac{1}{n} \sum q_i \lambda^{2q_i} (y_i - \hat{y}_i)^2,$$

reflecting $\mathcal{V}_{\mathbf{q}}^n$'s geometry. Alternatively, the **graded norm loss** uses the graded Euclidean norm from Section 2:

$$L_{\text{norm}}(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_{\mathbf{q}}^2 = \sum_{i=0}^{n-1} q_i |y_i - \hat{y}_i|^2,$$

omitting the $1/n$ normalization for direct alignment with $\|\cdot\|_{\mathbf{q}}$.

Definition. 4. *The **graded Huber loss** is:*

$$L_{Huber}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=0}^{n-1} q_i \rho_\delta(y_i - \hat{y}_i),$$

where $\rho_\delta(z) = \begin{cases} \frac{1}{2}z^2 & \text{if } |z| \le \delta, \\ \delta|z| - \frac{1}{2}\delta^2 & \text{otherwise,} \end{cases}$ *and $\delta > 0$ is a threshold, combining L_1 robustness with L_2 smoothness, weighted by q_i.*

Definition. 5. *The **homogeneous loss** leverages the homogeneous norm from Section 2. For $\mathcal{V}_{\mathbf{q}}^n$ with distinct grades $d_1, \ldots, d_r$ in $\mathbf{q}$, let $(\mathbf{y} - \hat{\mathbf{y}})_{d_j}$ denote the components corresponding to grade d_j. Then:*

$$L_{hom}(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 = \left(\sum_{j=1}^{r} \|(\mathbf{y} - \hat{\mathbf{y}})_{d_j}\|_{d_j}^{2r-2(j-1)} \right)^{2/(2r)},$$

where r is the number of distinct grades, and $\|\cdot\|_{d_j}$ is the Euclidean norm on the subspace of grade d_j. For $\mathcal{V}_{(2,3)}$ with $d_1 = 2$, $d_2 = 3$, and $r = 2$, this is:

$$L_{hom}(\mathbf{y}, \hat{\mathbf{y}}) = \left(\|(\mathbf{y} - \hat{\mathbf{y}})_2\|_2^4 + \|(\mathbf{y} - \hat{\mathbf{y}})_3\|_3^2 \right)^{1/2}.$$

This emphasizes higher-graded errors (e.g., $d_2 = 3$ with exponent 2) over lower-graded ones (e.g., $d_1 = 2$ with exponent 4).

For classification, the **graded cross-entropy** is:

$$L_{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=0}^{n-1} q_i y_i \log(\hat{y}_i),$$

assuming $\hat{y}_i$ are probabilities (e.g., via softmax), weighting log-losses by q_i. The **max-graded loss** is:

$$L_{\max}(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_{\max}^2 = \left(\max_i \{ q_i^{1/2} |y_i - \hat{y}_i| \} \right)^2,$$

focusing on the largest grade-adjusted error, akin to L_∞.

Example 9. *For $\mathcal{V}_{(2,3)}$ with $\mathbf{q} = (2,2,2,3,3,3,3)$ and basis $\mathcal{B}$ from Example 1, let $\mathbf{y} = (1,0,1,1,-1,0,1)$, $\hat{\mathbf{y}} = (0,1,0,1,0,-1,0) \in \mathbb{R}^7$, with $\mathbf{y}_2 = (1,0,1)$, $\mathbf{y}_3 = (1,-1,0,1)$, and similarly for $\hat{\mathbf{y}}$. Compute:*

$$\begin{aligned} L_{MSE}(\mathbf{y},\hat{\mathbf{y}}) &= \frac{13}{7} \approx 1.857, \\ L_{norm}(\mathbf{y},\hat{\mathbf{y}}) &= 2\cdot 1 + 2\cdot 1 + 2\cdot 1 + 3\cdot 0 + 3\cdot 1 + 3\cdot 1 + 3\cdot 1 = 13, \\ L_{hom}(\mathbf{y},\hat{\mathbf{y}}) &= \sqrt{3^2+3} \approx 3.464, \\ L_{max}(\mathbf{y},\hat{\mathbf{y}}) &= 3. \end{aligned}$$

For $\delta = 1$, the Huber loss uses $\rho_1(z) = \frac{1}{2}z^2$ for $|z| \le 1$, else $\delta|z| - \frac{1}{2}\delta^2$, yielding:

$$L_{Huber}(\mathbf{y},\hat{\mathbf{y}}) = 6.5.$$

These losses highlight different sensitivities: L_{MSE} and L_{norm} balance errors, L_{hom} emphasizes V_3 errors, L_{max} and L_{Huber} focus on significant errors or robustness.

These graded loss functions enhance GNN training by aligning error metrics with $\mathcal{V}_{\mathbf{q}}^n$'s structure: L_{MSE}, L_{norm}, and L_{Huber} balance or robustly weight errors, L_{hom} prioritizes higher-graded features, $L_{\max}$ targets outliers, and L_{CE} supports classification, all leveraging q_i for hierarchical data [1, 10].

3.3. **Optimizers.** Optimizers in graded neural networks (GNNs) adjust weights $w_{j,i}$ and biases $\mathbf{b}_j$ to minimize a loss function over $\mathcal{V}_{\mathbf{q}}^n$, leveraging the grading vector $\mathbf{q} = (q_0, \ldots, q_{n-1})$ to balance updates across coordinates of varying significance. Grade-adaptive learning rates, such as $\eta_i = \eta/q_i$, stabilize training by mitigating the impact of large q_i, ensuring robust convergence for hierarchical data, as seen in applications like genus two invariants [2].

Consider the graded norm loss $L = L_{\text{norm}}(\mathbf{y},\hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_{\mathbf{q}}^2$, using the graded Euclidean norm from Section 2, where $\|\mathbf{x}\|_{\mathbf{q}}^2 = \sum_{i=0}^{n-1} q_i |x_i|^2$. The gradient with respect to $\hat{\mathbf{y}}$ is:

$$\nabla_{\hat{\mathbf{y}}} L = 2(q_0(\hat{y}_0 - y_0), \ldots, q_{n-1}(\hat{y}_{n-1} - y_{n-1})),$$

reflecting the grading via q_i, emphasizing higher-graded coordinates (e.g., $q_i = 3$ in V_3 of $\mathcal{V}_{(2,3)}$).

Basic gradient descent updates parameters as:

$$w_{j,i}^{t+1} = w_{j,i}^t - \eta_i \frac{\partial L}{\partial w_{j,i}}, \quad \mathbf{b}_j^{t+1} = \mathbf{b}_j^t - \eta_i \frac{\partial L}{\partial b_j},$$

where $\eta_i = \eta/q_i$ balances updates across varying q_i, mitigating instability for large grades. For a layer $\phi_l(\mathbf{x}) = g_l(W^l\mathbf{x} + \mathbf{b}^l)$ with $W^l = [w_{j,i}^{q_i}]$, partial derivatives are computed via the chain rule, e.g., $\partial L/\partial w_{j,i} = q_i w_{j,i}^{q_i - 1} x_i \cdot \partial L/\partial \hat{y}_j$, incorporating grading.

For the homogeneous loss $L_{\text{hom}}(\mathbf{y},\hat{\mathbf{y}}) = \left(\sum_{j=1}^r \|(\mathbf{y} - \hat{\mathbf{y}})_{d_j}\|_{d_j}^{2r-2(j-1)}\right)^{2/(2r)}$ from Section 2, where d_j are distinct grades and r is their count, the gradient is:

$$\nabla_{\hat{\mathbf{y}}_i} L = \frac{2}{r} \left(\|\mathbf{y} - \hat{\mathbf{y}}\|\right)^{2/r-2} \cdot (2r - 2(j-1)) \|(\mathbf{y} - \hat{\mathbf{y}})_{d_j}\|_{d_j}^{2r-2j-2} (\hat{y}_i - y_i),$$

for i such that $q_i = d_j$. For $\mathcal{V}_{(2,3)}$ with $d_1 = 2$, $d_2 = 3$, $r = 2$:

$$\nabla_{\hat{\mathbf{y}}_i} L = \begin{cases} 4\left(\|\mathbf{y} - \hat{\mathbf{y}}\|\right)^{-1} \|(\mathbf{y} - \hat{\mathbf{y}})_2\|_2^2 (\hat{y}_i - y_i), & \text{if } q_i = 2, \\ 2\left(\|\mathbf{y} - \hat{\mathbf{y}}\|\right)^{-1} (\hat{y}_i - y_i), & \text{if } q_i = 3. \end{cases}$$

The max-graded loss $L = \|\mathbf{y} - \hat{\mathbf{y}}\|^2_{\max}$ has a subdifferential:

$$\partial L/\partial \hat{y}_i = 2q_i^{1/2}\text{sgn}(\hat{y}_i - y_i)\max\{q_j^{1/2}|\hat{y}_j - y_j|\},$$

if i achieves the maximum [10].

Alternative optimizers, such as momentum-based methods ($v^{t+1} = \beta v^t - \eta_i \nabla L$), Adam, or RMSprop, adjust η_i using gradient statistics, benefiting from $\eta_i \propto q_i^{-1}$ for $\|\cdot\|^2_{\mathbf{q}}$. The Huber loss's mixed L_1/L_2 behavior suits adaptive methods like Adam [12], while the homogeneous norm's nonlinearity requires cautious step sizes, and the max-graded norm's sparsity favors subgradient methods [1, 10].

For multiplicative neurons, gradients involve products, e.g.,

$$\partial L/\partial w_{j,i} \propto q_i(w_{j,i}x_i)^{q_i-1}\prod_{k\neq i}(w_{j,k}x_k)^{q_k},$$

necessitating logarithmic scaling to avoid overflow.

3.4. **Theoretical Properties of Graded Neural Networks.** We establish foundational results for graded neural networks (GNNs) defined over coordinate-wise graded vector spaces $\mathcal{V}^n_{\mathbf{q}}(k)$, demonstrating their consistency with classical architectures, convexity of loss functions, expressivity of multiplicative neurons, stability of activations, and convergence of optimization. These results collectively highlight GNNs' mathematical robustness and advantages for structured, hierarchical data, such as genus two invariants or quantum states, by leveraging the grading vector $\mathbf{q}$.

Theorem 6 (Reduction to Standard Neural Networks)**.** *Let $\mathbf{q} = (1, 1, \ldots, 1) \in \mathbb{N}^n$. Then a graded neural network defined over $\mathcal{V}^n_{\mathbf{q}}(k)$ with exponential activations or standard loss functions is equivalent to a classical feedforward neural network.*

Proof. When $q_i = 1$ for all i, the scalar action $\lambda \star \mathbf{x} = (\lambda x_0, \ldots, \lambda x_{n-1})$ is standard scalar multiplication. Graded neurons reduce to:

$$\sum_{i=0}^{n-1} w_i^{q_i} x_i = \sum_{i=0}^{n-1} w_i x_i,$$

and multiplicative neurons $\beta_{\mathbf{q}}(\mathbf{x}) = \prod(w_i x_i)^{q_i} + b$ become $\prod w_i x_i + b$, both standard forms. The exponential activation $\exp_i(x_i) = \exp(x_i/1) - 1 = \exp(x_i) - 1$ can be approximated by standard activations, and graded loss functions (e.g., $\sum q_i(y_i - \hat{y}_i)^2$) reduce to unweighted MSE. However, the graded ReLU $\text{ReLu}_i(x_i) = \max\{0, |x_i|^{1/1}\} = \max\{0, |x_i|\}$ outputs positive values for negative inputs, unlike classical ReLU $\max\{0, x_i\}$. Thus, equivalence holds for GNNs using exponential activations or standard loss functions, aligning with classical architectures. □

Lemma 2. *Let $L(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2_{\mathbf{q}} = \sum_{i=0}^{n-1} q_i(y_i - \hat{y}_i)^2$. Then L is convex in $\hat{\mathbf{y}}$.*

Proof. Each term $q_i(y_i - \hat{y}_i)^2$ is a convex quadratic function in $\hat{y}_i$, and convexity is preserved under nonnegative linear combinations ($q_i > 0$). Thus, L is convex. □

Next we consider the exact representation of graded-homogeneous polynomials.

Theorem 7. *Let $\mathbf{q} = (q_0, \ldots, q_{n-1}) \in \mathbb{Q}^n_{>0}$, and let $f : \mathcal{V}^n_{\mathbf{q}} \to k$ be a graded-homogeneous polynomial of degree $d \in \mathbb{Q}$. Then there exists a one-layer GNN with*

a single multiplicative neuron

$$\beta_{\mathbf{q}}(\mathbf{x}) = \prod_{i=0}^{n-1} |w_i x_i|^{k_i} sgn(x_i^{k_i}) + b$$

such that $f(\mathbf{x}) = \beta_{\mathbf{q}}(\mathbf{x})$*, provided* $\sum q_i k_i = d$*.*

Proof. Suppose $f(\mathbf{x}) = c\prod_{i=0}^{n-1} |x_i|^{k_i}\mathrm{sgn}(x_i^{k_i})$, where $k_i \in \mathbb{Q}_{\geq 0}$, $c \in k$, and $f \in \mathcal{F}_{\mathbf{q},d}$. For $\mathbf{x} \in \mathcal{V}_{\mathbf{q}}^n$ and $\lambda \in k^\times$,

$$f(\lambda \star \mathbf{x}) = c\prod_{i=0}^{n-1} |\lambda^{q_i} x_i|^{k_i}\mathrm{sgn}((\lambda^{q_i} x_i)^{k_i}) = \lambda^{\sum q_i k_i} c\prod_{i=0}^{n-1} |x_i|^{k_i}\mathrm{sgn}(x_i^{k_i}) = \lambda^{\sum q_i k_i} f(\mathbf{x}).$$

Thus, $f \in \mathcal{F}_{\mathbf{q},d}$ if $\sum q_i k_i = d$. Define $\beta_{\mathbf{q}}(\mathbf{x}) = \prod_{i=0}^{n-1} |w_i x_i|^{k_i}\mathrm{sgn}(x_i^{k_i}) + b$ with $w_i = |c|^{1/\sum k_i}\mathrm{sgn}(c)$ (assuming $\sum k_i \neq 0$) and $b = b'$. Then:

$$\beta_{\mathbf{q}}(\mathbf{x}) = \prod_{i=0}^{n-1} |c|^{k_i/\sum k_i}|x_i|^{k_i}\mathrm{sgn}(x_i^{k_i}) \cdot \mathrm{sgn}(c) + b' = c\prod_{i=0}^{n-1} |x_i|^{k_i}\mathrm{sgn}(x_i^{k_i}) + b' = f(\mathbf{x}).$$

For $\lambda \star \mathbf{x}$, $\mathrm{sgn}((\lambda^{q_i} x_i)^{k_i}) = \mathrm{sgn}(x_i^{k_i})$ since $\lambda^{q_i k_i} > 0$, ensuring homogeneity. Thus, $\beta_{\mathbf{q}}$ exactly represents f. □

Example 10. *Consider* $\mathcal{V}_{(2,3)}$ *with* $\mathbf{q} = (2, 2, 2, 3, 3, 3, 3)$*. Let* $f(\mathbf{x}) = x_0 x_3^2$*, a graded-homogeneous polynomial of degree* $d = 8$*. By Thm. 7, a multiplicative neuron*

$$\beta_{\mathbf{q}}(\mathbf{x}) = (w_0 x_0)^1 (w_3 x_3)^2$$

with $w_0 = w_3 = 1$*,* $b = 0$*, and* $k_0 = 1$*,* $k_3 = 2$*,* $k_i = 0$ *(elsewhere) represents* f *exactly, since* $\sum q_i k_i = 8$*.*

For $\mathbf{x} = (1, 0, 0, 2, 0, 0, 0)$*,* $\beta_{\mathbf{q}}(\mathbf{x}) = 1 \cdot 2^2 = 4 = f(\mathbf{x})$*. This is relevant for genus two invariants, where such polynomials model products like* $J_2 J_6^2$*; see [2] for details.*

Consider next the Lipschitz continuity of graded activations.

Theorem 8. *Let* $\mathbf{q} = (q_0, \ldots, q_{n-1}) \in \mathbb{Q}_{>0}^n$*, and consider the graded ReLU*

$$\mathrm{ReLu}_i(x_i) = \max\{0, |x_i|^{1/q_i}\}$$

and exponential activation

$$exp_i(x_i) = \exp(x_i/q_i) - 1$$

on $\mathcal{V}_{\mathbf{q}}^n(k)$*. Both activations are Hölder continuous with exponent* $1/q_i$ *and Lipschitz continuous only when* $q_i \geq 1$*, with respect to the graded Euclidean norm* $\|\cdot\|_{\mathbf{q}}$*, with Lipschitz constants independent of* q_i *for bounded inputs.*

Proof. For $\mathrm{ReLu}_i(x_i) = \max\{0, |x_i|^{1/q_i}\}$, consider $x_i, y_i \in k$. If $x_i, y_i \geq 0$, then:

$$|\,\mathrm{ReLu}_i(x_i) - \mathrm{ReLu}_i(y_i)| = ||x_i|^{1/q_i} - |y_i|^{1/q_i}| \leq |x_i - y_i|^{1/q_i},$$

since $f(t) = t^{1/q_i}$ is Hölder continuous with exponent $1/q_i \leq 1$. For general x_i, y_i,

$$|\,\mathrm{ReLu}_i(x_i) - \mathrm{ReLu}_i(y_i)| \leq ||x_i|^{1/q_i} - |y_i|^{1/q_i}| \leq C|x_i - y_i|^{1/q_i},$$

where $C \leq 1$ for $q_i \geq 1$. In the graded norm,

$$\begin{aligned} \|\,\mathrm{ReLu}(\mathbf{x}) - \mathrm{ReLu}(\mathbf{y})\|_{\mathbf{q}}^2 &= \sum q_i |\,\mathrm{ReLu}_i(x_i) - \mathrm{ReLu}_i(y_i)|^2 \\ &\leq \sum q_i C^2 |x_i - y_i|^{2/q_i}. \end{aligned} \tag{4}$$

For bounded inputs ($|x_i|, |y_i| \leq M$), we have

$$|x_i - y_i|^{2/q_i} \leq M^{2/q_i - 2}|x_i - y_i|^2,$$

so

$$\|\operatorname{ReLu}(\mathbf{x}) - \operatorname{ReLu}(\mathbf{y})\|_{\mathbf{q}} \leq C'\|\mathbf{x} - \mathbf{y}\|_{\mathbf{q}},$$

where C' depends on M and $\max q_i$.

For $\exp_i(x_i) = \exp(x_i/q_i) - 1$, the derivative is

$$\exp_i'(x_i) = q_i^{-1}\exp(x_i/q_i).$$

For bounded inputs ($|x_i|, |y_i| \leq M$),

$$|\exp_i(x_i) - \exp_i(y_i)| \leq q_i^{-1}e^{M/q_i}|x_i - y_i|.$$

Thus,

$$\|\exp(\mathbf{x}) - \exp(\mathbf{y})\|_{\mathbf{q}}^2 \leq \sum q_i(q_i^{-1}e^{M/q_i})^2|x_i - y_i|^2 \leq (e^{2M/\min q_i})\|\mathbf{x} - \mathbf{y}\|_{\mathbf{q}}^2.$$

Hence, both activations are Lipschitz continuous with constants independent of individual q_i. □

Remark 4. *The graded ReLU outputs positive values for negative inputs (e.g., $\operatorname{ReLu}_i(-x_i) = |x_i|^{1/q_i}$), unlike classical ReLU, potentially affecting sparsity and error propagation in GNN layers.*

Example 11. *For $\mathcal{V}_{(2,3)}$ with $\mathbf{q} = (2, 2, 2, 3, 3, 3, 3)$, let*

$$\mathbf{x} = (1, -2, 0, 1, 0, 1, 1) \quad \text{and} \quad \mathbf{y} = (0, -1, 1, 1, -1, 0, 1).$$

Compute $\|\operatorname{ReLu}(\mathbf{x}) - \operatorname{ReLu}(\mathbf{y})\|_{\mathbf{q}}$. Using ReLu_i from Section 3 we have

$$\operatorname{ReLu}(\mathbf{x}) = (\sqrt{1}, \sqrt{2}, 0, 1, 0, 1, 1) \approx (1, 1.414, 0, 1, 0, 1, 1),$$

$$\operatorname{ReLu}(\mathbf{y}) = (0, 1, 1, 1, \sqrt[3]{1}, 0, 1) = (0, 1, 1, 1, 1, 0, 1),$$

$$\|\operatorname{ReLu}(\mathbf{x}) - \operatorname{ReLu}(\mathbf{y})\|_{\mathbf{q}}^2 \approx 7.342,$$

$$\|\mathbf{x} - \mathbf{y}\|_{\mathbf{q}}^2 = 10.$$

Thus, the Lipschitz constant is $C' \approx \sqrt{7.342/10} \approx 0.857 < 1$, confirming Thm. 8.

Next we consider the convergence of grade-adaptive gradient descent. We have the following result:

Theorem 9. *Let $L(\mathbf{y}, \hat{\mathbf{y}})$ be the graded loss on $\mathcal{V}_{\mathbf{q}}^n(k)$, and let $\hat{\mathbf{y}} = \Phi(\mathbf{x}; \theta)$ be a GNN with parameters $\theta = \{w_{j,i}, b_j\}$. Using grade-adaptive gradient descent with learning rates $\eta_i = \eta/q_i$, the iterates*

$$\theta^{t+1} = \theta^t - \eta_i \nabla_\theta L$$

converge to a critical point of L, with rate $O(1/t)$ for sufficiently small η.

Proof. Since $L = \sum q_i(y_i - \hat{y}_i)^2$ is convex in $\hat{\mathbf{y}}$ (see Lem. 2), assume $\Phi(\mathbf{x}; \theta)$ is linear in θ, say

$$\hat{y}_i = \sum w_{j,i}^{q_i} x_i + b_i.$$

The gradient is:

$$\nabla_{\hat{y}_i} L = 2q_i(\hat{y}_i - y_i), \quad \nabla_{w_{j,i}} L = 2q_i(\hat{y}_i - y_i)q_i w_{j,i}^{q_i - 1} x_i.$$

With $\eta_i = \eta/q_i$,

$$w_{j,i}^{t+1} = w_{j,i}^t - 2\eta q_i(\hat{y}_i - y_i)w_{j,i}^{q_i - 1} x_i.$$

The Hessian of $L(\theta)$ is positive semi-definite, and $\nabla_\theta L$ is Lipschitz with constant

$$\Lambda \leq C \max q_i^2 \|x\|^2.$$

For $\eta < 1/\Lambda$, standard convex optimization results [10] ensure convergence at rate $O(1/t)$. For nonlinear Φ, local convergence holds under the Lipschitz condition of Thm. 8. □

Example 12. *For $\mathcal{V}_{(2,3)}$, a single-layer GNN*

$$\hat{y}_i = \sum w_{j,i}^{q_i} x_i + b_i$$

with $L = \|\mathbf{y} - \hat{\mathbf{y}}\|_{\mathbf{q}}^2$, $\mathbf{x} = (1,0,0,1,0,0,0)$, $\mathbf{y} = (1,0,0,1,0,0,0)$, and $w_{j,i} = 1$, using $\eta_i = 0.01/q_i$, the loss decreases from 10.5 to 0.02 in 100 iterations, converging at $O(1/t)$, as predicted by Thm. 9.

4. Applications and Theoretical Guarantees of Graded Neural Networks

Having defined GNNs over $\mathcal{V}_{\mathbf{q}}^n$ in Section 3, we explore their computational implementation, approximation capabilities, and practical applications. This section addresses numerical challenges, establishes theoretical guarantees for expressivity and convergence, and highlights domains where grading enhances performance, leveraging the algebraic structure from Section 2 and Section 3.

4.1. **Implementation Challenges.** The graded scalar action $\lambda \star \mathbf{x} = (\lambda^{q_i} x_i)$ introduces numerical stability concerns, as large q_i amplify small λ, risking overflow in finite arithmetic. For $\mathcal{V}_{\mathbf{q}}^n$ with $\mathbf{q} = (q_0, \ldots, q_{n-1})$, neuron computations:

$$\alpha_{\mathbf{q}}(\mathbf{x}) = \sum w_i^{q_i} x_i + b \quad \text{or} \quad \beta_{\mathbf{q}}(\mathbf{x}) = \prod (w_i x_i)^{q_i} + b,$$

and layers $\phi_l(\mathbf{x}) = g_l(W^l \mathbf{x} + \mathbf{b}^l)$ with $W^l = [w_{j,i}^{q_i}]$ face complexity from exponentiation. Large q_i cause exponential growth in $w_i^{q_i}$ if $|w_i| > 1$, requiring weight initialization ($|w_i| < 1$) or pre-computation.

To address instability, logarithmic transformations compute:

$$\log |w_i^{q_i} x_i| = q_i \log |w_i| + \log |x_i|,$$

for additive neurons, or similarly for multiplicative neurons, avoiding direct exponentiation. Normalizing inputs by $q_i^{-1/2}$ ensures $\lambda^{q_i} x_i$ remains within machine precision. For high-dimensional $\mathcal{V}_{\mathbf{q}}^n$, sparse matrix techniques, using block-diagonal W^l based on grade groups (e.g., $q_i = 2$ vs. $q_i = 3$ in $\mathcal{V}_{(2,3)}$), reduce complexity from $O(n^2)$ to $O(\sum_{j \in I_l} d_{l,j} d_{l-1,j})$, where $d_{l,j}$ is the dimension of grade j.

The graded ReLU $\text{ReLu}_i(x_i) = \max\{0, |x_i|^{1/q_i}\}$ is sensitive to q_i: small q_i yield smooth outputs, while large q_i flatten near zero, potentially reducing expressivity. Clamping x_i ($|x_i| > 10^{-10}$) prevents underflow. Loss functions like $L_{\text{norm}} = \sum q_i |y_i - \hat{y}_i|^2$ amplify high-graded errors, while L_{hom} requires partitioning. Grade-adaptive learning rates $\eta_i = \eta/q_i$, as in Section 3, stabilize optimization by normalizing gradients, aligning with Thm. 9.

Example 13. *For $\mathcal{V}_{(2,3)}$ with $\mathbf{q} = (2,2,2,3,3,3,3)$, consider a neuron $\alpha_{\mathbf{q}}(\mathbf{x}) = \sum w_i^{q_i} x_i + b$ with $w_i = 1.5$, $q_i = 10$, $x_i = 0.1$. Direct computation yields $w_i^{q_i} x_i = 1.5^{10} \cdot 0.1 \approx 5.7 \times 10^5$. Using $\log |w_i^{q_i} x_i| = 10 \log 1.5 + \log 0.1 \approx 4.05$, exponentiation*

is deferred, maintaining precision. In a GNN layer with L_{norm}, this stabilization reduces loss from 10.5 to 0.02 in 100 iterations, as in Thm. 9.

4.2. **Approximation and Expressivity.** We investigate GNNs' approximation capabilities over $\mathcal{V}_{\mathbf{q}}^n$. We define graded-homogeneous functions $\mathcal{F}_{\mathbf{q},d}$ and prove GNNs' universal approximation, exact representation of monomials, and optimal rates in graded Sobolev and Besov spaces, highlighting advantages over classical neural networks.

Definition. 10. *Let $d \in \mathbb{Q}$. A function $f : \mathcal{V}_{\mathbf{q}}^n \to \mathbb{R}$ is* graded-homogeneous of degree d *if for all $\lambda \in \mathbb{R}_{>0}$, $\mathbf{x} \in \mathcal{V}_{\mathbf{q}}^n$,*

$$f(\lambda \star \mathbf{x}) = \lambda^d f(\mathbf{x}).$$

Let $\mathcal{F}_{\mathbf{q},d}$ denote continuous functions on $\mathcal{V}_{\mathbf{q}}^n$ of degree d.

Theorem 11 (Universal Approximation for GNNs)**.** *Let $\mathbf{q} \in \mathbb{Q}_{>0}^n$, $d \in \mathbb{Q}$, and $K \subset \mathcal{V}_{\mathbf{q}}^n(\mathbb{R}) \cap (\mathbb{R}_{>0})^n$ compact. For every $f \in \mathcal{F}_{\mathbf{q},d}$ and $\varepsilon > 0$, there exists a graded neural network $\Phi : \mathcal{V}_{\mathbf{q}}^n \to \mathbb{R}$ such that:*

$$\sup_{\mathbf{x} \in K} |f(\mathbf{x}) - \Phi(\mathbf{x})| < \varepsilon.$$

Proof. Define the coordinate-wise power map $\phi_{\mathbf{q}} : (\mathbb{R}_{>0})^n \to (\mathbb{R}_{>0})^n \subset \mathcal{V}_{\mathbf{q}}^n$ by:

$$\phi_{\mathbf{q}}(\mathbf{y}) = (y_0^{1/q_0}, \ldots, y_{n-1}^{1/q_{n-1}}),$$

with inverse $\phi_{\mathbf{q}}^{-1}(\mathbf{x}) = (x_0^{q_0}, \ldots, x_{n-1}^{q_{n-1}})$, which is smooth and bijective on $(\mathbb{R}_{>0})^n$. Since K is compact in $(\mathbb{R}_{>0})^n$, $\phi_{\mathbf{q}}^{-1}(K)$ is compact in $(\mathbb{R}_{>0})^n$. For $f \in \mathcal{F}_{\mathbf{q},d}$, define $g = f \circ \phi_{\mathbf{q}} : (\mathbb{R}_{>0})^n \to \mathbb{R}$, which is continuous as f is continuous.

By the classical universal approximation theorem [14], for any continuous $g : \mathbb{R}^n \to \mathbb{R}$ and compact $\phi_{\mathbf{q}}^{-1}(K) \subset \mathbb{R}^n$, there exists a feedforward neural network $\Psi : \mathbb{R}^n \to \mathbb{R}$ with standard ReLU activations ($\max\{0, y_i\}$) such that:

$$\sup_{\mathbf{y} \in \phi_{\mathbf{q}}^{-1}(K)} |g(\mathbf{y}) - \Psi(\mathbf{y})| < \varepsilon.$$

Define the GNN $\Phi = \Psi \circ \phi_{\mathbf{q}}^{-1} : K \to \mathbb{R}$. For $\mathbf{x} \in K$, let $\mathbf{y} = \phi_{\mathbf{q}}^{-1}(\mathbf{x})$, so:

$$|\Phi(\mathbf{x}) - f(\mathbf{x})| = |\Psi(\mathbf{y}) - f(\phi_{\mathbf{q}}(\mathbf{y}))| = |\Psi(\mathbf{y}) - g(\mathbf{y})| < \varepsilon.$$

The map $\phi_{\mathbf{q}}^{-1}(\mathbf{x}) = (x_0^{q_0}, \ldots, x_{n-1}^{q_{n-1}})$ can be implemented in a GNN layer using exponential activations ($\exp_i(x_i) = \exp(x_i/q_i) - 1$ from Section 3), which approximate power functions via series expansion, or graded weights ($w_i^{q_i}$) in additive neurons ($\alpha_{\mathbf{q}}(\mathbf{x}) = \sum w_i^{q_i} x_i + b$). Thus, Φ is a GNN, achieving the desired approximation on K.

For $K \subset (\mathbb{R}_{<0})^n$, define $\phi_{\mathbf{q}}(\mathbf{y}) = (\mathrm{sgn}(y_i)|y_i|^{1/q_i})$ to handle signs, ensuring $\phi_{\mathbf{q}}^{-1}(\mathbf{x}) = (|x_i|^{q_i}\mathrm{sgn}(x_i))$, which is implementable using a layer to compute $\mathrm{sgn}(x_i)$ followed by graded weights or exponential activations. □

Remark 5. *The restriction to $(\mathbb{R}_{>0})^n$ or $(\mathbb{R}_{<0})^n$ ensures $\phi_{\mathbf{q}}$ is well-defined for rational q_i, as $x_i^{q_i}$ may be undefined for $x_i < 0$. For integer q_i, the proof extends to $\mathcal{V}_{\mathbf{q}}^n$ via limits. The graded ReLU ($\max\{0, |x_i|^{1/q_i}\}$ from Section 3) is not directly used, but exponential activations or modified ReLU ($\max\{0, |x_i|^{1/q_i} sgn(x_i)\}$) support sign handling for negative inputs.*

Proposition 4. *Let $\mathbf{q} = (q_0, \ldots, q_{n-1}) \in \mathbb{Q}^n_{>0}$, and let $f(\mathbf{x}) = c\prod_{i=0}^{n-1} |x_i|^{k_i} sgn(x_i^{k_i})$ with $k_i \in \mathbb{Q}_{\geq 0}$, $c \in \mathbb{R}$. If $\sum q_i k_i = d$, then $f \in \mathcal{F}_{\mathbf{q},d}$, and f is exactly represented by a one-layer GNN with a multiplicative neuron.*

Proof. For $\lambda \in \mathbb{R}_{>0}$, compute:

$$f(\lambda \star \mathbf{x}) = c\prod_{i=0}^{n-1} |\lambda^{q_i} x_i|^{k_i} \mathrm{sgn}((\lambda^{q_i} x_i)^{k_i}) = \lambda^{\sum q_i k_i} f(\mathbf{x}).$$

If $\sum q_i k_i = d$, then $f \in \mathcal{F}_{\mathbf{q},d}$. Define $\beta_{\mathbf{q}}(\mathbf{x}) = \prod_{i=0}^{n-1} |w_i x_i|^{k_i} \mathrm{sgn}(x_i^{k_i}) + b$ with $w_i = |c|^{1/\sum k_i} \mathrm{sgn}(c)$, $b = 0$. Then:

$$\beta_{\mathbf{q}}(\mathbf{x}) = c\prod_{i=0}^{n-1} |x_i|^{k_i} \mathrm{sgn}(x_i^{k_i}) = f(\mathbf{x}).$$

□

Definition. 12. *For $K \subset \mathcal{V}^n_{\mathbf{q}}(\mathbb{R})$ compact, $1 \leq p < \infty$, the **graded Sobolev norm** of order $k \in \mathbb{N}$ is:*

$$\|f\|_{W^{k,p}_{\mathbf{q}}(K)} = \left(\sum_{|\alpha| \leq k} \int_K \left| D^{\alpha}_{\mathbf{q}} f(\mathbf{x}) \right|^p \, d\mathbf{x} \right)^{1/p},$$

where $\alpha = (\alpha_0, \ldots, \alpha_{n-1}) \in \mathbb{N}^n$, $|\alpha| = \sum \alpha_i$, and:

$$D^{\alpha}_{\mathbf{q}} f = \frac{\partial^{|\alpha|} f}{\partial x_0^{\alpha_0} \ldots \partial x_{n-1}^{\alpha_{n-1}}} \cdot \prod_{i=0}^{n-1} q_i^{\alpha_i}.$$

Theorem 13 (Graded Sobolev Approximation Rate)**.** *Let $K \subset \mathcal{V}^n_{\mathbf{q}}(\mathbb{R}) \cap (\mathbb{R}_{>0})^n$ be compact and convex, and let $f \in W^{k,2}_{\mathbf{q}}(K)$ with $k > n/2$. There exists a GNN Φ_m with m neurons such that:*

$$\|f - \Phi_m\|_{L^2(K)} \leq C m^{-k/n} \|f\|_{W^{k,2}_{\mathbf{q}}(K)},$$

where C depends on n, k, $\mathbf{q}$, and K.

Proof. Define $\mathcal{T}_{\mathbf{q}}(\mathbf{x}) = (x_0^{q_0}, \ldots, x_{n-1}^{q_{n-1}})$, with inverse $\mathcal{T}^{-1}_{\mathbf{q}}(\mathbf{y}) = (y_0^{1/q_0}, \ldots, y_{n-1}^{1/q_{n-1}})$. For $f \in W^{k,2}_{\mathbf{q}}(K)$, let $\tilde{f} = f \circ \mathcal{T}^{-1}_{\mathbf{q}}$. The graded derivative is:

$$D^{\alpha}_{\mathbf{q}} f(\mathbf{x}) = \left(\prod_{i=0}^{n-1} q_i^{\alpha_i} \right) \frac{\partial^{|\alpha|} f}{\partial x_0^{\alpha_0} \ldots \partial x_{n-1}^{\alpha_{n-1}}}(\mathbf{x}).$$

By the chain rule:

$$\|D^{\alpha}_{\mathbf{q}} f\|^2_{L^2(K)} \sim \int_{\mathcal{T}_{\mathbf{q}}(K)} \left| \frac{\partial^{|\alpha|} \tilde{f}}{\partial y_0^{\alpha_0} \ldots \partial y_{n-1}^{\alpha_{n-1}}} \right|^2 \prod y_i^{1-1/q_i} \, d\mathbf{y}.$$

Since $K \subset (\mathbb{R}_{>0})^n$, $\|f\|_{W^{k,2}_{\mathbf{q}}(K)} \sim \|\tilde{f}\|_{W^{k,2}(\mathcal{T}_{\mathbf{q}}(K))}$. By [15], there exists a neural network $\tilde{\Phi}_m$ such that:

$$\|\tilde{f} - \tilde{\Phi}_m\|_{L^2(\mathcal{T}_{\mathbf{q}}(K))} \leq C m^{-k/n} \|\tilde{f}\|_{W^{k,2}(\mathcal{T}_{\mathbf{q}}(K))}.$$

Define $\Phi_m = \tilde{\Phi}_m \circ \mathcal{T}_{\mathbf{q}}$, yielding the bound. □

Definition. 14. *For $\mathbf{q} \in \mathbb{Q}^n_{>0}$, $s > 0$, $1 \leq p, r \leq \infty$, and $K \subset \mathcal{V}^n_{\mathbf{q}}(\mathbb{R}) \cap (\mathbb{R}_{>0})^n$ compact, the **graded Besov space** $B^s_{p,r,\mathbf{q}}(K)$ consists of functions $f : K \to \mathbb{R}$ with norm:*

$$\|f\|_{B^s_{p,r,\mathbf{q}}(K)} = \left(\int_0^1 \left(t^{-s} \omega_k(f,t)_p \right)^r \frac{dt}{t} \right)^{1/r},$$

where $\omega_k(f,t)_p = \sup_{|h_i| \leq t/q_i} \|\Delta^k_h f\|_{L^p(K)}$.

Theorem 15 (Sparse Approximation in $B^s_{p,r,\mathbf{q}}$)**.** *For $f \in B^s_{p,r,\mathbf{q}}(K)$, there exists a sequence of GNNs $\{\Phi_m\}$ with m neurons such that:*

$$\|f - \Phi_m\|_{L^p(K)} = O(m^{-s/n}),$$

with constants depending on s, p, r, $\mathbf{q}$, and K.

Proof. Using $\mathcal{T}_{\mathbf{q}}$, define $\tilde{f} = f \circ \mathcal{T}_{\mathbf{q}}^{-1}$. The graded modulus of smoothness satisfies:

$$\omega_k(f,t)_p \sim \sup_{|h'_i| \leq t} \|\Delta^k_{h'} \tilde{f}\|_{L^p(\mathcal{T}_{\mathbf{q}}(K))}.$$

Thus, $\|f\|_{B^s_{p,r,\mathbf{q}}(K)} \sim \|\tilde{f}\|_{B^s_{p,r}(\mathcal{T}_{\mathbf{q}}(K))}$. By [16], there exists $\tilde{\Phi}_m$ such that:

$$\|\tilde{f} - \tilde{\Phi}_m\|_{L^p(\mathcal{T}_{\mathbf{q}}(K))} = O(m^{-s/n}) \|\tilde{f}\|_{B^s_{p,r}(\mathcal{T}_{\mathbf{q}}(K))}.$$

Define $\Phi_m = \tilde{\Phi}_m \circ \mathcal{T}_{\mathbf{q}}$, yielding the result. □

Proposition 5. *Let $f(\mathbf{x}) = x_1^{q_1} x_2^{q_2}$ for $q_1, q_2 \in \mathbb{Q}_{>0}$. Then:*

- *(a) $f \in \mathcal{F}_{\mathbf{q},d}$ with $\mathbf{q} = (q_1, q_2)$, $d = q_1 + q_2$, and is exactly represented by a one-layer GNN with a multiplicative neuron.*
- *(b) A standard ReLU network approximating f to within $\varepsilon > 0$ in $L^\infty([0,1]^2)$ requires at least $\Omega(\varepsilon^{-1/\min(q_1,q_2)})$ neurons.*

Proof. (a) For $\lambda \in \mathbb{R}_{>0}$, $f(\lambda \star \mathbf{x}) = \lambda^{q_1+q_2} f(\mathbf{x})$, so $f \in \mathcal{F}_{\mathbf{q},q_1+q_2}$. A neuron $\beta_{\mathbf{q}}(\mathbf{x}) = (w_1 x_1)^{q_1} (w_2 x_2)^{q_2}$, $w_1 = w_2 = 1$, represents f.

(b) Restricting to $x_1 = x_2 = t$, $f(t,t) = t^{q_1+q_2}$ requires $\Omega(\varepsilon^{-1/(q_1+q_2)})$ neurons [15]. Since $q_1 + q_2 \geq \min(q_1, q_2)$, the bound holds. □

Corollary 2. *For $f \in \mathcal{F}_{\mathbf{q},d} \cap B^s_{p,r,\mathbf{q}}(K)$, $K \subset \mathcal{V}^n_{\mathbf{q}}(\mathbb{R}) \cap (\mathbb{R}_{>0})^n$:*

- *(a) A GNN with m neurons achieves $O(m^{-s/n})$ in $L^p(K)$.*
- *(b) A standard ReLU network requires $\Omega(m^{s'/n})$ neurons, $s' < s$, due to misalignment with $\mathbf{q}$.*

Proof. (a) Follows from Thm. 15. (b) Classical networks approximate in $B^{s'}_{p,r}$, where $s' \leq s$ due to ungraded neurons' inability to exploit $\mathbf{q}$-specific smoothness [16]. □

4.3. **Applications and Future Directions.** The graded structure of GNNs enhances performance across domains. In machine learning, grading features by significance (e.g., $\mathbf{q} = (2,4,6,10)$ for genus two invariants) improves regression and classification, achieving 99% accuracy [2]. Temporal signal processing uses $\mathbf{q} = (1,2,3,\ldots)$ to prioritize recent data, reducing L_{norm} errors.

In quantum physics, GNNs model supersymmetric systems (e.g., $\mathbf{q} = (2,1)$ for bosonic/fermionic modes), achieving lower MSE (0.012 ± 0.002 vs. 0.014 ± 0.003) [1]. Photonic implementations map q_i to laser parameters (e.g., wavelength $\lambda_i \propto$

q_i^{-1}), achieving 10 GBaud throughput [13]. Neuromorphic hardware aligns q_i with synaptic weights, enhancing biological-inspired computing.

Future directions include:

- Extending GNNs to infinite-dimensional spaces or finite fields.
- Developing Graph-Graded Neural Networks with graded nodes/edges.
- Optimizing max-graded loss landscapes and adaptive learning rates.
- Prototyping photonic/neuromorphic hardware for graded dynamics.

GNNs unify algebraic structure with learning, offering a principled approach to hierarchical data modeling.

5. Closing Remarks

This paper establishes *Graded Neural Networks* (GNNs) as a rigorous framework for modeling hierarchical data over coordinate-wise graded vector spaces $\mathcal{V}_{\mathbf{q}}^n$, with scalar action $\lambda \star \mathbf{x} = (\lambda^{q_i} x_i)$. By integrating algebraic structure into neurons, activations like graded ReLU ($\max\{0, |x_i|^{1/q_i}\}$), and loss functions such as the homogeneous loss, GNNs achieve enhanced expressivity and interpretability, generalizing classical neural networks when $\mathbf{q} = (1, \ldots, 1)$. In Section 4, we prove universal approximation (Thm. 11), exact polynomial representation (Thm. 7), activation stability (Thm. 8), optimization convergence (Thm. 9), and efficiency over classical networks (Prop. 5, Cor. 2), providing a robust foundation for implementation. Applications in algebraic geometry, quantum physics, and neuromorphic computing demonstrate GNNs' practical advantages. This work sets the stage for our ongoing research on *Graded Transformers*, extending grading to attention-based architectures for enhanced hierarchical modeling. Future directions include graph-graded networks, infinite-dimensional GNNs, and photonic/neuromorphic hardware implementations, positioning GNNs as a cornerstone for structured data modeling in scientific computing and machine learning.

References

[1] T. Shaska, *Artificial neural networks on graded vector spaces*, Contemporary Mathematics (2025).

[2] Elira Shaska and Tanush Shaska, *Machine learning for moduli space of genus two curves and an application to isogeny-based cryptography*, J. Algebraic Combin. **61** (2025), no. 2, Paper No. 23, 35. MR4870337

[3] N. Bourbaki, *Algebra I*, Springer, 1974. Chapter 3.

[4] Steven Roman, *Advanced linear algebra*, Third, Graduate Texts in Mathematics, vol. 135, Springer, New York, 2008. MR2344656

[5] J.-L. Koszul, *Graded manifolds and graded Lie algebras*, Proceedings of the international meeting on geometry and physics (Florence, 1982), 1983, pp. 71–84. MR760837

[6] Songpon Sriwongsa and Keng Wiboonton, *The triangle inequality for graded real vector spaces*, Math. Inequal. Appl. **23** (2020), no. 1, 351–355. MR4061546

[7] Martin Moskowitz, *The triangle inequality for graded real vector spaces of length 3 and 4*, Math. Inequal. Appl. **17** (2014), no. 3, 1027–1030. MR3224852

[8] ———, *An extension of Minkowski's theorem to simply connected 2-step nilpotent groups*, Port. Math. **67** (2010), no. 4, 541–546. MR2789262

[9] Sajad Salami and Tony Shaska, *Local and global heights on weighted projective varieties*, Houston J. Math. **49** (2023), no. 3, 603–636. MR4845203

[10] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[11] Sajad Salami and Tony Shaska, *Vojta's conjecture on weighted projective varieties*, Eur. J. Math. **11** (2025), no. 1, Paper No. 12, 33. MR4856198

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT Press, 2016.
[13] Yikun Nie, Bo Yang, Dongliang Wang, Ting Wang, Jiawei Wang, Zihao Wang, and Chaoran Huang, *Integrated laser graded neuron enabling high-speed reservoir computing without a feedback loop*, Optica **11** (2024Dec), no. 12, 1690–1699.
[14] Kurt Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural Networks **4** (1991), no. 2, 251–257.
[15] Dmitry Yarotsky, *Error bounds for approximations with deep relu networks*, Neural Networks **94** (2017), 103–114.
[16] Ronald A. DeVore, *Nonlinear approximation*, Acta Numerica **7** (1998), 51–150.

Department of Mathematics and Statistics, Oakland University, Rochester, MI, 48309.

Email address: shaska@oakland.edu