GRADED TRANSFORMERS: A SYMBOLIC-GEOMETRIC APPROACH TO STRUCTURED LEARNING

TONY SHASKA

To my parents!
This work is for them—unbroken, unbowed.

Abstract. Transformers are highly effective for sequence modeling, yet they struggle to capture hierarchical structure without extensive training, limiting both efficiency and interpretability. We introduce the *Graded Transformer*, a novel architecture that integrates the dynamic learning capabilities of transformers with the algebraic inductive biases of Graded Neural Networks (GNNs). The core mechanism is a grading transformation $G_{\mathbf{w},\lambda}$, parameterized by a grading tuple and scaling factor, which prioritizes features according to structural importance. This enables the model to better handle hierarchical tasks across domains such as algebraic geometry (e.g., polynomial systems), physics (e.g., turbulent flows), natural language processing (e.g., dependency parsing), and biological sequence analysis (e.g., genomic variant prediction). We formalize the model, analyze its architecture, and prove key properties—including universal approximation, attention rank enhancement, reduced sample complexity, and robustness to noise. A graded loss function supports effective training and deployment. This framework offers a principled approach to interpretable and efficient sequence modeling across structured scientific and linguistic domains.

1. Introduction

Sequence modeling underpins modern machine learning, enabling breakthroughs in natural language processing (NLP), time-series analysis, and biological sequence analysis by capturing long-range dependencies across tokens. The transformer architecture has revolutionized this field through self-attention, dynamically prioritizing token interactions to achieve state-of-the-art performance across tasks like machine translation, physical simulations, and genomic analysis; see [32]. However, transformers face significant challenges with hierarchical or graded data structures, prevalent in domains such as algebraic geometry (e.g., polynomial degrees of varying importance), physics (e.g., multi-scale phenomena with dominant energy levels), NLP (e.g., syntactic heads in parse trees), and biology (e.g., genetic sequences with critical regulatory regions). Their unstructured attention mechanisms require extensive training data to uncover domain-specific patterns, leading to high sample complexity, increased computational costs, and limited interpretability when hierarchical relationships are known a priori; see [36] for details.

Efforts to address these limitations, such as structured attention mechanisms and graph neural networks, often introduce relational biases at the cost of transformer flexibility or necessitate complex preprocessing as in [32]. Graded Neural Networks (GNNs), introduced in [21], offer a compelling alternative, embedding algebraic biases into neural architectures to prioritize features based on domain knowledge; see

[15] for further details. Grounded in graded vector spaces, GNNs assign numerical grades to features, enabling static prioritization that enhances efficiency and interpretability for tasks like photonic signal processing or genetic sequence analysis and many other applications in mathematical research.

This paper introduces the Graded Transformer, a novel extension that synergizes the dynamic, context-aware learning of transformers with the static, algebraically motivated biases of GNNs. By incorporating grading transformations

$$G_{\mathbf{w},\lambda} = \operatorname{diag}(\lambda^{q_0}, \ldots, \lambda^{q_{d-1}})$$

(cf. Definition 2.2), the Graded Transformer embeds hierarchical priors into sequence modeling, emphasizing critical features or positions without relying solely on data-driven attention. This approach pursues three primary objectives:

(1) **Feature Prioritization**: Highlighting significant features, such as high-degree polynomial terms in algebraic geometry, key phrases in NLP, or regulatory regions in genomics, to reduce dependence on large datasets.
(2) **Computational Efficiency**: Leveraging structural priors to lower sample complexity, enabling faster convergence for hierarchical tasks like physical system modeling or low-resource language processing.
(3) **Interpretability**: Encoding domain knowledge transparently via grading tuples, making the model's behavior predictable and explainable, particularly for scientific applications.

The Graded Transformer is uniquely suited to domains with intrinsic hierarchical structures, offering a versatile framework for applications in algebraic geometry, physics, NLP, biological sequence analysis, and cross-domain transfer learning. The paper is structured to develop this framework comprehensively.

Section 2 establishes the algebraic foundations of graded vector spaces and GNNs, formalizing feature prioritization mechanisms.

Section 4 defines the Graded Transformer, proving its universal approximation (Thm. 4.6), attention rank enhancement (Prop. 4.7), and robustness properties. Section 5 details the architecture, integrating grading across inputs, positional encodings, attention, feed-forward layers, and outputs, with stability guarantees.

Section 6 explores training and optimization strategies, ensuring practical applicability via graded loss functions. Section 7 delineates domain-specific applications, from polynomial systems to genomic sequences.

Section 8 synthesizes contributions and outlines future directions, including empirical validation and architectural extensions. This introduction frames the motivation and significance of the Graded Transformer, paving the way for a rigorous exploration of its theoretical and practical advancements.

Throughout this paper we assume familiarity with graded vector spaces and neural networks in the level of [15]. By $\mathbb{F}$ we denote a field, V a vector space over $\mathbb{F}$, and $\mathbf{x} \in V$ a column vector.

2. Preliminaries

This section lays the mathematical groundwork for the Graded Transformer by introducing graded vector spaces and Graded Neural Networks (GNNs), which extend the framework of artificial neural networks on graded vector spaces [15].

2.1. **Graded Vector Spaces.** A graded vector space equips subspaces with numerical grades, enabling differential scaling of components to reflect their relative importance. This algebraic structure underpins the Graded Transformer by providing a mechanism to prioritize features in machine learning tasks, such as high-degree terms in algebraic geometry or critical tokens in natural language processing.

Definition 2.1 (Graded Vector Space)**.** *Let $\mathbb{F}$ be a field, and let V be a vector space over $\mathbb{F}$ with basis $\mathcal{B} = \{e_0, \ldots, e_{d-1}\}$. A* ***graded vector space*** *is a direct sum decomposition*

$$V = \bigoplus_{i=0}^{d-1} V_i,$$

where $V_i = \mathbb{F}e_i$ is the one-dimensional subspace spanned by e_i, assigned a grade $q_i \in \mathbb{R}$. The grades form the ***grading tuple*** *$\mathbf{w} = (q_0, \ldots, q_{d-1})$. A vector $\mathbf{x} \in V$ is expressed as*

$$\mathbf{x} = \sum_{i=0}^{d-1} x_i e_i, \quad x_i \in \mathbb{F},$$

with the component $x_i e_i \in V_i$ having grade q_i.

In neural network applications, we set $\mathbb{F} = \mathbb{R}$, and real-valued grades q_i allow continuous prioritization of features, such as high-frequency components in photonic signals [15]. Unlike algebraic settings, where grades are often integers (e.g., graded rings [14]), real grades enhance flexibility for machine learning.

The grading tuple induces a linear transformation that scales vector components by their grades, formalizing feature prioritization in the Graded Transformer.

Definition 2.2 (Grading Transformation)**.** *Let $V = \bigoplus_{i=0}^{d-1} \mathbb{R}e_i$ be a graded vector space over $\mathbb{R}$ with grading tuple $\mathbf{w} = (q_0, \ldots, q_{d-1})$, and let $\lambda > 0$. The* ***grading transformation*** *is the linear operator $G_{\mathbf{w},\lambda} : V \to V$, represented in the basis $\mathcal{B} = \{e_i\}_{i=0}^{d-1}$ by the diagonal matrix*

$$G_{\mathbf{w},\lambda} = \operatorname{diag}(\lambda^{q_0}, \ldots, \lambda^{q_{d-1}}).$$

For a vector $\mathbf{x} = [x_0, \ldots, x_{d-1}]^t \in V$,

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

To study the properties of $G_{\mathbf{w},\lambda}$, we employ standard tools from linear algebra. For a vector $\mathbf{x} \in \mathbb{R}^d$, the **Euclidean norm** is

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=0}^{d-1} x_i^2}.$$

For a linear operator $A : \mathbb{R}^d \to \mathbb{R}^d$, the **spectral norm** is

$$\|A\|_2 = \sup_{\mathbf{x} \neq 0} \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2},$$

the largest singular value of A. For a diagonal matrix $A = \operatorname{diag}(a_0, \ldots, a_{d-1})$, this is $\|A\|_2 = \max_i |a_i|$. The spectral norm bounds the operator's scaling effect, crucial for numerical stability in neural networks. A mapping $f : \mathbb{R}^d \to \mathbb{R}^d$ is **Lipschitz continuous** with constant $L \geq 0$ if

$$\|f(\mathbf{x}) - f(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, where the smallest such L is the Lipschitz constant. Lipschitz continuity ensures controlled sensitivity to input changes, vital for robust sequence modeling.

Lemma 2.3. *Let $q_{\max}$ denote the maximum of weights $q_0, \ldots, q_{d-1}$. The grading transformation $G_{\mathbf{w},\lambda}$ satisfies:*

i) Invertibility*: For $\lambda > 0$, $G_{\mathbf{w},\lambda}$ is invertible.*

ii) Scaling*: For $\mu \in \mathbb{R}$,*

$$G_{\mathbf{w},\lambda}(\mu \mathbf{x}) = \mu \cdot G_{\mathbf{w},\lambda}(\mathbf{x}).$$

iii) Norm Bound*: For $\mathbf{x} \in \mathbb{R}^d$, assuming $\lambda > 1$ and $q_i \geq 0$,*

$$\|G_{\mathbf{w},\lambda}(\mathbf{x})\|_2 \leq \lambda^{q_{\max}} \cdot \|\mathbf{x}\|_2.$$

iv) Spectral Norm*: The eigenvalues of $G_{\mathbf{w},\lambda}$ are λ^{q_i}, with spectral norm $\|G_{\mathbf{w},\lambda}\|_2 = \lambda^{q_{\max}}$.*

v) Lipschitz Continuity*: The mapping*

$$\mathbf{x} \mapsto G_{\mathbf{w},\lambda}(\mathbf{x})$$

is Lipschitz continuous with constant $\lambda^{q_{\max}}$.

Proof. i) Since $G_{\mathbf{w},\lambda} = \operatorname{diag}(\lambda^{q_0}, \ldots, \lambda^{q_{d-1}})$ and $\lambda > 0$, each $\lambda^{q_i} > 0$. The inverse is

$$G_{\mathbf{w},\lambda}^{-1} = \operatorname{diag}(\lambda^{-q_0}, \ldots, \lambda^{-q_{d-1}}),$$

satisfying $G_{\mathbf{w},\lambda} G_{\mathbf{w},\lambda}^{-1} = I$.

ii) For $\mu \in \mathbb{R}$, we have

$$G_{\mathbf{w},\lambda}(\mu \mathbf{x}) = (\lambda^{q_0}(\mu x_0), \ldots, \lambda^{q_{d-1}}(\mu x_{d-1})) = \mu G_{\mathbf{w},\lambda}(\mathbf{x}).$$

iii) Compute $\|G_{\mathbf{w},\lambda}(\mathbf{x})\|_2$ as follows

$$\|G_{\mathbf{w},\lambda}(\mathbf{x})\|_2 = \sqrt{\sum_{i=0}^{d-1} (\lambda^{q_i} x_i)^2} = \sqrt{\sum_{i=0}^{d-1} \lambda^{2q_i} x_i^2}.$$

For $\lambda > 1$ and $q_i \geq 0$, $\lambda^{q_i} \leq \lambda^{q_{\max}}$, so

$$\sum_{i=0}^{d-1} \lambda^{2q_i} x_i^2 \leq \lambda^{2q_{\max}} \sum_{i=0}^{d-1} x_i^2 = \lambda^{2q_{\max}} \|\mathbf{x}\|_2^2.$$

Thus, $\|G_{\mathbf{w},\lambda}(\mathbf{x})\|_2 \leq \lambda^{q_{\max}} \|\mathbf{x}\|_2$.

iv) As $G_{\mathbf{w},\lambda}$ is diagonal, its eigenvalues are λ^{q_i}. The spectral norm is

$$\|G_{\mathbf{w},\lambda}\|_2 = \max_i |\lambda^{q_i}| = \lambda^{q_{\max}},$$

since $\lambda^{q_i} > 0$ for all $i = 0, \ldots, d-1$.

v) For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$\|G_{\mathbf{w},\lambda}(\mathbf{x}) - G_{\mathbf{w},\lambda}(\mathbf{y})\|_2 = \|G_{\mathbf{w},\lambda}(\mathbf{x} - \mathbf{y})\|_2 \leq \lambda^{q_{\max}} \|\mathbf{x} - \mathbf{y}\|_2,$$

by part iii, confirming the Lipschitz constant. □

2.2. **Graded Neural Networks.** Graded Neural Networks (GNNs) extend traditional neural networks by embedding graded vector spaces, introducing algebraic biases to prioritize features based on domain knowledge. Introduced in [21], this framework addresses the challenge of modeling hierarchical data, reducing sample complexity and enhancing interpretability for tasks such as algebraic geometry and sequence processing, paving the way for the Graded Transformer.

Definition 2.4 (Graded Neural Network)**.** *A **Graded Neural Network** (GNN) is a neural network whose input space, hidden layers, or output space are graded vector spaces over* $\mathbb{R}$*.*

For an input $\mathbf{x} \in \mathbb{R}^d$*, a GNN layer applies a grading transformation* $G_{\mathbf{w},\lambda}$*, with grading tuple* $\mathbf{w} = (q_0, \ldots, q_{d-1})$ *and* $\lambda > 0$*, via*

$$\mathbf{y} = \sigma(WG_{\mathbf{w},\lambda}(\mathbf{x}) + \mathbf{b}),$$

where $W \in \mathbb{R}^{m\times d}$ *is a weight matrix,* $\mathbf{b} \in \mathbb{R}^m$ *is a bias, and* $\sigma : \mathbb{R} \to \mathbb{R}$ *is an activation function (e.g., ReLU) applied element-wise.*

Alternatively, grading may be applied post-activation, as

$$\mathbf{y} = G_{\mathbf{w}',\lambda'}\sigma(W\mathbf{x} + \mathbf{b}),$$

with distinct grading tuple $\mathbf{w}'$ *and scalar* $\lambda' > 0$*. In multi-layer GNNs, each layer may use unique* $\mathbf{w}_l, \lambda_l$*.*

Graded neural networks were first defined in [21]. We now establish stability properties of GNN layers, leveraging the grading transformation's boundedness from Lem. 2.3.

Lemma 2.5 (Lipschitz Continuity of GNN Layer)**.** *Let*

$$f(\mathbf{x}) = \sigma(WG_{\mathbf{w},\lambda}(\mathbf{x}) + \mathbf{b})$$

be a GNN layer, where σ *is Lipschitz continuous with constant* L_σ*, and* $\|W\|_2$ *denotes the spectral norm of* W*. Then* f *is Lipschitz continuous with constant at most* $L_\sigma\|W\|_2\lambda^{q_{\max}}$*.*

Proof. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, compute

$$\|f(\mathbf{x}) - f(\mathbf{y})\|_2 = \|\sigma(WG_{\mathbf{w},\lambda}(\mathbf{x}) + \mathbf{b}) - \sigma(WG_{\mathbf{w},\lambda}(\mathbf{y}) + \mathbf{b})\|_2.$$

Since σ is Lipschitz with constant L_σ,

$$\|\sigma(WG_{\mathbf{w},\lambda}(\mathbf{x}) + \mathbf{b}) - \sigma(WG_{\mathbf{w},\lambda}(\mathbf{y}) + \mathbf{b})\|_2 \le L_\sigma\|WG_{\mathbf{w},\lambda}(\mathbf{x}) - WG_{\mathbf{w},\lambda}(\mathbf{y})\|_2.$$

The operator norm yields

$$\|WG_{\mathbf{w},\lambda}(\mathbf{x}) - WG_{\mathbf{w},\lambda}(\mathbf{y})\|_2 = \|WG_{\mathbf{w},\lambda}(\mathbf{x} - \mathbf{y})\|_2 \le \|W\|_2\|G_{\mathbf{w},\lambda}(\mathbf{x} - \mathbf{y})\|_2.$$

By Lem. 2.3 (part iii), assuming $\lambda > 1$ and $q_i \ge 0$,

$$\|G_{\mathbf{w},\lambda}(\mathbf{x} - \mathbf{y})\|_2 \le \lambda^{q_{\max}}\|\mathbf{x} - \mathbf{y}\|_2.$$

Combining these, we obtain

$$\|f(\mathbf{x}) - f(\mathbf{y})\|_2 \le L_\sigma\|W\|_2\lambda^{q_{\max}}\|\mathbf{x} - \mathbf{y}\|_2,$$

confirming the Lipschitz constant. □

Proposition 2.6 (Multi-Layer GNN Stability)**.** *A multi-layer GNN with L layers, each defined by $f_l(\mathbf{x}) = \sigma_l(W_l G_{\mathbf{w}_l,\lambda_l}\mathbf{x} + \mathbf{b}_l)$, is Lipschitz continuous with constant at most*

$$\prod_{l=1}^{L} L_{\sigma_l}\|W_l\|_2 \lambda_l^{q_{l,\max}},$$

where $q_{l,\max} = \max_{i=0,\ldots,d-1} q_{l,i}$.

Proof. Let $f = f_L \circ \cdots \circ f_1$ be the composition of the L layers. By Lem. 2.5, each layer f_l is Lipschitz continuous with constant $L_{\sigma_l}\|W_l\|_2\lambda_l^{q_{l,\max}}$. For inputs $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the Lipschitz property of compositions gives

$$\|f(\mathbf{x}) - f(\mathbf{y})\|_2 \leq \prod_{l=1}^{L} \|f_l(\mathbf{z}_{l-1}) - f_l(\mathbf{z}'_{l-1})\|_2 \leq \prod_{l=1}^{L} L_{\sigma_l}\|W_l\|_2\lambda_l^{q_{l,\max}}\|\mathbf{x} - \mathbf{y}\|_2,$$

where $\mathbf{z}_l = f_l \circ \cdots \circ f_1(\mathbf{x})$, $\mathbf{z}'_l = f_l \circ \cdots \circ f_1(\mathbf{y})$, and the product follows from chaining the Lipschitz constants. □

Next example illustrates how a GNN could process a photonic signal, such as the intensity of an optical field in communication or imaging systems; see [13] for some of the fundamentals of photonics.

Example 2.7. *Let $\mathbf{x} \in \mathbb{R}^d$ represent the signal in the frequency domain, obtained via a discrete Fourier transform, where x_i is the amplitude of the i-th frequency component, $i = 0, \ldots, d-1$.*

In tasks like signal classification (e.g., identifying modulation types) or denoising (e.g., enhancing optical images), high-frequency components often encode critical features, such as rapid modulations or edges, but are prone to noise; see [12] for further details.

Consider a GNN layer

$$\mathbf{y} = \sigma(W G_{\mathbf{w},\lambda}\mathbf{x} + \mathbf{b}),$$

with grading tuple $\mathbf{w} = (q_0, \ldots, q_{d-1})$, $q_i = ci$, $c > 0$, and $\lambda > 1$. The grading transformation

$$G_{\mathbf{w},\lambda}\mathbf{x} = (\lambda^{q_0}x_0, \ldots, \lambda^{q_{d-1}}x_{d-1})$$

amplifies higher frequencies.

For instance, if $d = 3$, $\mathbf{x} = [1, 0.5, 0.1]$, $c = 0.1$, $\lambda = 2$, then

$$G_{\mathbf{w},\lambda}\mathbf{x} \approx [1, 0.535, 0.116],$$

emphasizing the highest frequency. This pre-emphasis simplifies the weight matrix W's role, as critical features are scaled prior to learning. By Lem. 2.5, the layer's stability ensures robust prioritization, enhancing efficiency for frequency-dependent tasks; see [1, 5] for further details.

The algebraic structure of GNNs, combining static grading transformations with dynamic neural network learning, provides a robust framework for hierarchical feature prioritization. The stability guarantees of Lem. 2.5 and Prop. 2.6 enable GNNs to serve as a foundation for the Graded Transformer's context-aware sequence modeling, as developed next.

3. Transformers

Let $V = \{1, 2, \ldots, |V|\}$ be a finite vocabulary of tokens, with $|V| \in \mathbb{N}$. For $d \in \mathbb{N}$, let $\mathbb{R}^d$ be the embedding space for token representations. Define V^n as the set of sequences of length n over V, and let:

$$\mathcal{S}_{\text{in}} = \bigcup_{n=1}^{n_{\max}} V^n, \quad \mathcal{S}_{\text{out}} = \bigcup_{m=1}^{m_{\max}} V^m,$$

be the spaces of input and output sequences, where $n_{\max}, m_{\max} \in \mathbb{N}$ are maximum lengths. Denote $\text{Mat}_{m,n}(\mathbb{R})$ as the set of $m \times n$ matrices with real entries. A sequence $(t_1, \ldots, t_n) \in V^n$ is represented as a matrix in $\text{Mat}_{n,d}(\mathbb{R})$ after embedding and positional encoding, as defined below.

A **transformer** is a function:

$$\mathcal{T}_\theta : \mathcal{S}_{\text{in}} \to \mathcal{S}_{\text{out}},$$

parameterized by a collection of learnable parameters θ, mapping an input sequence $t = (t_1, \ldots, t_n) \in V^n$ to an output sequence $s = (s_1, \ldots, s_m) \in V^m$, with $m \leq m_{\max}$ determined by the generation process. The parameters θ include embedding matrices, attention weights, feed-forward weights, and normalization parameters, specified in each component.

The transformer comprises three main stages: input embedding with positional encoding, an encoder, and an autoregressive decoder. The architecture of a transformer is displayed below.

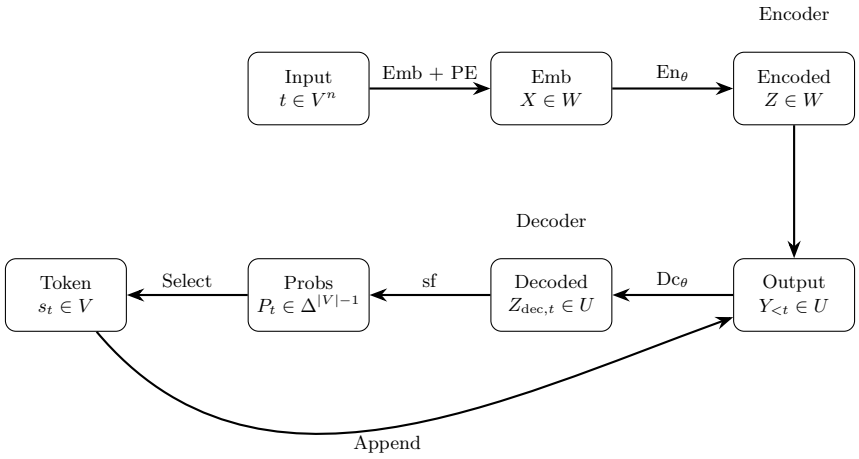

FIGURE 1. Transformer architecture mapping $t \in V^n$ to $s \in V^m$, where $W = \text{Mat}_{n,d}(\mathbb{R})$, $U = \text{Mat}_{t,d}(\mathbb{R})$.

3.1. Input Embedding and Positional Encoding.

Define the embedding matrix $W_e \in \text{Mat}_{|V|,d}(\mathbb{R})$. For a token $t_i \in V$, its embedding is:

$$\mathbf{x}_i = W_e \cdot \text{onehot}(t_i) \in \mathbb{R}^d,$$

where $\text{onehot}(t_i) \in \{0, 1\}^{|V|}$ has a 1 at index t_i.

To incorporate positional information, define the **positional encoding** function

$$PE : \{1, \dots, n_{\max}\} \to \mathbb{R}^d$$

given by

$$PE(i)_k = \begin{cases} \sin\left(\frac{i}{10000^{k/d}}\right) & \text{if } k \text{ is even,} \\ \cos\left(\frac{i}{10000^{(k-1)/d}}\right) & \text{if } k \text{ is odd,} \end{cases}$$

for $k = 0, \dots, d-1$. For an input sequence $t = (t_1, \dots, t_n)$, the embedded input matrix is:

$$X = [\mathbf{x}_1 + PE(1), \dots, \mathbf{x}_n + PE(n)]^T \in \mathrm{Mat}_{n,d}(\mathbb{R}),$$

where the rows are $\mathbf{x}_i + PE(i)$.

3.2. **Encoder.** The encoder is a function:

$$\mathrm{En}_\theta : \mathrm{Mat}_{n,d}(\mathbb{R}) \to \mathrm{Mat}_{n,d}(\mathbb{R}),$$

transforming X into a contextualized representation $Z = \mathrm{En}_\theta(X)$. It consists of $L \in \mathbb{N}$ layers, each applying multi-head self-attention, a feed-forward network, residual connections, and layer normalization.

Let $\mathrm{Mat}_{m,n}(\mathbb{R})$ denote the vector space of $m \times n$ matrices with real entries, and $\mathrm{Mat}_n(\mathbb{R}) = \mathrm{Mat}_{n,n}(\mathbb{R})$. For a matrix $M = [a_{i,j}] \in \mathrm{Mat}_{m,n}(\mathbb{R})$, the **softmax function** is defined as:

$$\mathrm{sf} : \mathrm{Mat}_{m,n}(\mathbb{R}) \to \left\{ P \in \mathrm{Mat}_{m,n}(\mathbb{R}) \mid p_{i,j} \geq 0, \sum_{j=1}^{n} p_{i,j} = 1 \text{ for all } i \right\},$$

$$\mathrm{sf}(M)_{i,j} = \frac{\exp(a_{i,j})}{\sum_{k=1}^{n} \exp(a_{i,k})}.$$

This function is well-defined, differentiable, and normalizes each row into a probability distribution.

Definition 3.1 (Multi-Head Self-Attention)**.** *Let $h \in \mathbb{N}$ be the number of attention heads, and $d_k = d/h \in \mathbb{N}$. For layer $l = 1, \dots, L$, head $i = 1, \dots, h$, define parameter matrices $W_{Q,l,i}, W_{K,l,i}, W_{V,l,i} \in \mathrm{Mat}_{d,d_k}(\mathbb{R})$. For $X \in \mathrm{Mat}_{n,d}(\mathbb{R})$:*

$$Q_i = XW_{Q,l,i}, \quad K_i = XW_{K,l,i}, \quad V_i = XW_{V,l,i} \in \mathrm{Mat}_{n,d_k}(\mathbb{R}).$$

The attention for head i is:

$$A_{l,i}(X) = \mathrm{sf}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \in \mathrm{Mat}_{n,d_k}(\mathbb{R}).$$

The multi-head attention is:

$$\mathrm{Mh}_l(X) = \mathit{Concat}(A_{l,1}(X), \dots, A_{l,h}(X))W_{O,l} \in \mathrm{Mat}_{n,d}(\mathbb{R}),$$

with $W_{O,l} \in \mathrm{Mat}_{hd_k,d}(\mathbb{R})$.

Definition 3.2 (Feed-Forward Network)**.** *The feed-forward network*

$$\mathit{FFN}_l : \mathrm{Mat}_{n,d}(\mathbb{R}) \to \mathrm{Mat}_{n,d}(\mathbb{R})$$

is applied row-wise. For a row $\mathbf{z} \in \mathbb{R}^d$:

$$\mathit{FFN}_l(\mathbf{z}) = W_{2,l} \cdot \mathit{ReLU}(W_{1,l}\mathbf{z} + \mathbf{b}_{1,l}) + \mathbf{b}_{2,l},$$

where $W_{1,l} \in \mathrm{Mat}_{d,d_f}(\mathbb{R})$, $W_{2,l} \in \mathrm{Mat}_{d_f,d}(\mathbb{R})$, $\mathbf{b}_{1,l} \in \mathbb{R}^{d_f}$, $\mathbf{b}_{2,l} \in \mathbb{R}^d$, $d_f \in \mathbb{N}$, and $\mathit{ReLU}(x) = \max(0, x)$.

Definition 3.3 (Layer Normalization)**.** *Layer normalization*

$$\mathrm{Ln} : \mathrm{Mat}_{n,d}(\mathbb{R}) \to \mathrm{Mat}_{n,d}(\mathbb{R}) \tag{1}$$

is applied row-wise. For a row $\mathbf{z} \in \mathbb{R}^d$*:*

$$\mathrm{Ln}(\mathbf{z}) = \frac{\mathbf{z} - \mu}{\sqrt{\sigma^2 + \epsilon}} \cdot \boldsymbol{\gamma} + \boldsymbol{\beta},$$

where

$$\mu = \frac{1}{d}\sum_{k=1}^{d} z_k, \qquad \sigma^2 = \frac{1}{d}\sum_{k=1}^{d}(z_k - \mu)^2,$$

$\epsilon > 0$*, and* $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^d$*.*

The l-th encoder layer is:

$$\mathrm{EncLayer}_l(X) = \mathrm{Ln}(X' + \mathrm{FFN}_l(X')), \quad X' = \mathrm{Ln}(X + \mathrm{Mh}_l(X)).$$

The encoder is the composition:

$$\mathrm{En}_\theta(X) = \mathrm{EncLayer}_L \circ \mathrm{EncLayer}_{L-1} \circ \cdots \circ \mathrm{EncLayer}_1(X).$$

3.3. **Decoder.** The decoder is a function:

$$\mathrm{Dc}_\theta : \mathrm{Mat}_{t,d}(\mathbb{R}) \times \mathrm{Mat}_{n,d}(\mathbb{R}) \to \mathrm{Mat}_{t,d}(\mathbb{R}),$$

taking $Y_{<t} \in \mathrm{Mat}_{t,d}(\mathbb{R})$ (current output sequence) and $Z \in \mathrm{Mat}_{n,d}(\mathbb{R})$ (encoder output) to produce $Z_{\mathrm{dec}} = \mathrm{Dc}_\theta(Y_{<t}, Z)$. It consists of L layers, each with masked self-attention, cross-attention, and a feed-forward network.

Definition 3.4 (Masked Multi-Head Self-Attention)**.** *For* $Y \in \mathrm{Mat}_{t,d}(\mathbb{R})$*, compute:*

$$Q_i = YW_{Q,l,i}, \quad K_i = YW_{K,l,i}, \quad V_i = YW_{V,l,i} \in \mathrm{Mat}_{t,d_k}(\mathbb{R}),$$

with $W_{Q,l,i}, W_{K,l,i}, W_{V,l,i} \in \mathrm{Mat}_{d,d_k}(\mathbb{R})$*. The masked attention is:*

$$A_{l,i}(Y) = \mathrm{sf}\left(\frac{Q_i K_i^T}{\sqrt{d_k}} \cdot M\right) V_i,$$

where $M \in \mathrm{Mat}_{t,t}(\mathbb{R})$*,* $M_{i,j} = 1$ *if* $j \le i$*, and* $M_{i,j} = -\infty$ *if* $j > i$ *(implemented via masking before softmax). Then:*

$$\mathrm{Mmh}_l(Y) = \mathit{Concat}(A_{l,1}(Y), \ldots, A_{l,h}(Y))W_{O,l} \in \mathrm{Mat}_{t,d}(\mathbb{R}).$$

Definition 3.5 (Cross-Attention)**.** *For* $Y \in \mathrm{Mat}_{t,d}(\mathbb{R})$*,* $Z \in \mathrm{Mat}_{n,d}(\mathbb{R})$*:*

$$Q_i = YW_{Q,l,i} \in \mathrm{Mat}_{t,d_k}(\mathbb{R}), \quad K_i = ZW_{K,l,i}, \quad V_i = ZW_{V,l,i} \in \mathrm{Mat}_{n,d_k}(\mathbb{R}),$$

$$A_{l,i}(Y, Z) = \mathrm{sf}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \in \mathrm{Mat}_{t,d_k}(\mathbb{R}),$$

$$\mathrm{Ca}_l(Y, Z) = \mathit{Concat}(A_{l,1}(Y, Z), \ldots, A_{l,h}(Y, Z))W_{O,l} \in \mathrm{Mat}_{t,d}(\mathbb{R}).$$

The l-th decoder layer is:

$$\mathrm{DecLayer}_l(Y, Z) = \mathrm{Ln}(Y'' + \mathrm{FFN}_l(Y'')),$$

where:

$$Y' = \mathrm{Ln}(Y + \mathrm{Mmh}_l(Y)), \quad Y'' = \mathrm{Ln}(Y' + \mathrm{Ca}_l(Y', Z)).$$

The decoder is:

$$\mathrm{Dc}_\theta(Y, Z) = \mathrm{DecLayer}_L \circ \cdots \circ \mathrm{DecLayer}_1(Y, Z).$$

3.4. **Autoregressive Generation.** For an input $t \in V^n$, compute X as in Section 3.1 and $Z = \mathrm{En}_\theta(X)$. Initialize with a start token $s_0 \in V$, setting:

$$Y_{<1} = [\mathbf{y}_0 + PE(1)]^T \in \mathrm{Mat}_{1,d}(\mathbb{R}), \quad \mathbf{y}_0 = W_e \cdot \mathrm{onehot}(s_0).$$

For $t = 1, 2, \ldots$ we have:

1. Compute $Z_{\mathrm{dec},t} = \mathrm{Dc}_\theta(Y_{<t}, Z) \in \mathrm{Mat}_{t,d}(\mathbb{R})$.
2. Extract $\mathbf{z}_t = (Z_{\mathrm{dec},t})_{t,:} \in \mathbb{R}^d$.
3. Compute:
$$P_t = \mathrm{sf}(W_e^T \mathbf{z}_t) \in \Delta^{|V|-1},$$
where $\Delta^{|V|-1} = \{p \in \mathbb{R}^{|V|} \mid p_v \geq 0, \sum_v p_v = 1\}$, and $\mathrm{sf}(u)_v = \frac{\exp(u_v)}{\sum_w \exp(u_w)}$.
4. Select:
$$s_t = \min\{v \in V \mid P_{t,v} = \max_{u \in V} P_{t,u}\}.$$
5. Set $\mathbf{y}_t = W_e \cdot \mathrm{onehot}(s_t)$.
6. Update $Y_{<t+1} = [Y_{<t}^T, \mathbf{y}_t + PE(t+1)]^T \in \mathrm{Mat}_{t+1,d}(\mathbb{R})$.
7. Stop if $s_t = v_{\mathrm{eos}} \in V$ or $t = m_{\max}$.

The output is $\mathcal{T}_\theta(t) = (s_1, \ldots, s_m)$.

Theorem 3.6. *The transformer $\mathcal{T}_\theta : \mathcal{S}_{in} \to \mathcal{S}_{out}$ is a well-defined function.*

Proof. The embedding X is deterministic via W_e and PE. Each encoder layer (attention, FFN, normalization) is a composition of deterministic operations (matrix multiplication, softmax, ReLU). The encoder output $Z = \mathrm{En}_\theta(X)$ is unique. Decoder operations are similarly deterministic, and the min-rule in token selection ensures a unique s_t.

The generation stops when $s_t = v_{\mathrm{eos}}$ or $t = m_{\max}$, ensuring $m \leq m_{\max}$.

The autoregressive process produces a unique sequence $s = (s_1, \ldots, s_m)$, as each s_t depends deterministically on $Y_{<t}$ and Z, with $Y_{<t}$ uniquely determined by prior steps. Thus, $\mathcal{T}_\theta(t)$ assigns a unique output in $\mathcal{S}_{\mathrm{out}}$. □

This framework, while powerful for sequence modeling, lacks structural biases for hierarchical data, a limitation addressed by the Graded Transformer in Section 4.

Proposition 3.7 (Permutation-Equivariance)**.** *The self-attention operator*

$$A_{l,i}(X) = \mathrm{sf}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i$$

is permutation-equivariant. In other words, for any permutation matrix $P \in \mathrm{Mat}_n(\mathbb{R})$, if

$$Q_i' = PQ_i, \quad K_i' = PK_i, \quad V_i' = PV_i,$$

then

$$A_{l,i}(Q_i', K_i', V_i') = PA_{l,i}(Q_i, K_i, V_i).$$

Proof. Let $S = \frac{Q_i K_i^T}{\sqrt{d_k}}$ and

$$Q_i' K_i'^T = PQ_i K_i^T P^T = PSP^T, \quad \mathrm{sf}(PSP^T) = P\,\mathrm{sf}(S)P^T.$$

Then

$$A_{l,i}(Q_i', K_i', V_i') = \mathrm{sf}(PSP^T)(PV_i) = P\,\mathrm{sf}(S)V_i = PA_{l,i}(Q_i, K_i, V_i).$$

□

Proposition 3.8. *The complexity of a self-attention head $A_{l,i}(X)$ is $O(n^2 d_k)$, and for multi-head attention with h heads, $O(n^2 d)$.*

Proof. For one head: computing Q_i, K_i, V_i costs $O(ndd_k)$; $Q_i K_i^T$ costs $O(n^2 d_k)$; softmax costs $O(n^2)$; and $\mathrm{sf}(Q_i K_i^T/\sqrt{d_k})V_i$ costs $O(n^2 d_k)$. The total is dominated by $O(n^2 d_k)$. For h heads, with $d_k = d/h$, the complexity is $O(h \cdot n^2 d/h) = O(n^2 d)$. The feed-forward network costs $O(ndd_f)$. □

Proposition 3.9 (Scaling Factor)**.** *Let $Q, K \in \mathbb{R}^{n\times d_k}$, where each row $q_i \in \mathbb{R}^{d_k}$ and $k_j \in \mathbb{R}^{d_k}$ consists of independent entries sampled from $\mathcal{N}(0,1)$. Then for the dot-product attention matrix $S \in \mathbb{R}^{n\times n}$, defined by*

$$S_{i,j} = \frac{q_i \cdot k_j}{\sqrt{d_k}},$$

each entry has variance 1:

$$\mathrm{Var}(S_{i,j}) = 1.$$

Proof. Fix rows $q_i, k_j \in \mathbb{R}^{d_k}$, where $q_i = (Q_{i,1}, \ldots, Q_{i,d_k})$, $k_j = (K_{j,1}, \ldots, K_{j,d_k})$, with all $Q_{i,\ell}$, $K_{j,\ell}$ independently sampled from $\mathcal{N}(0,1)$. Then

$$q_i \cdot k_j = \sum_{\ell=1}^{d_k} Q_{i,\ell} K_{j,\ell}.$$

Each product $Q_{i,\ell} K_{j,\ell}$ has mean zero and variance 1, since the product of two independent $\mathcal{N}(0,1)$ variables has variance 1.

Since the d_k terms are independent, we have:

$$\mathrm{Var}(q_i \cdot k_j) = \sum_{\ell=1}^{d_k} \mathrm{Var}(Q_{i,\ell} K_{j,\ell}) = d_k.$$

After scaling:

$$\mathrm{Var}\left(\frac{q_i \cdot k_j}{\sqrt{d_k}}\right) = \frac{d_k}{d_k} = 1.$$

□

4. Graded Transformers

The Graded Transformer augments the transformer architecture by incorporating grading transformations to prioritize hierarchical features in sequence modeling, addressing the inefficiency of standard transformers in capturing structured patterns [32]. Extending the framework of graded vector spaces [15] and Graded Neural Networks (GNNs) [21], it enhances efficiency and interpretability for domains such as algebraic geometry (e.g., graded rings), physics (e.g., multi-scale phenomena), and natural language processing (e.g., syntactic hierarchies). This section defines the model, introduces its graded attention mechanism, and establishes its mathematical properties, laying the foundation for the architecture and training in Sections 5 and 6.

Let $(\mathbb{R}^d)^n$ denote the space of sequences $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, where $\mathbf{x}_i \in \mathbb{R}^d$. Given the grading transformation $G_{\mathbf{w},\lambda} = \mathrm{diag}(\lambda^{q_0}, \ldots, \lambda^{q_{d-1}})$, with grading tuple $\mathbf{w} =$

$(q_0, \ldots, q_{d-1})$, $q_i \geq 0$, $\lambda > 0$, and a standard transformer $\mathcal{T} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$, define the map

$$\phi_{\mathbf{w},\lambda} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n,$$
$$\phi_{\mathbf{w},\lambda}(X) = (G_{\mathbf{w},\lambda}\mathbf{x}_i)_{i=1}^n,$$

where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, and $G_{\mathbf{w},\lambda}$ is applied to each input vector $\mathbf{x}_i \in \mathbb{R}^d$.

A **Graded Transformer** $\mathcal{GT}_{\mathbf{w},\lambda} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ is defined as

$$Y = \mathcal{GT}_{\mathbf{w},\lambda}(X) = \mathcal{T}(\phi_{\mathbf{w},\lambda}(X)).$$

Proposition 4.1. *The Graded Transformer $\mathcal{GT}_{\mathbf{w},\lambda} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ is a well-defined map.*

Proof. The map $\phi_{\mathbf{w},\lambda} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ is well-defined, as $G_{\mathbf{w},\lambda} \in \mathrm{Mat}_d(\mathbb{R})$ is a linear transformation, and applying it to each $\mathbf{x}_i \in \mathbb{R}^d$ yields $G_{\mathbf{w},\lambda}\mathbf{x}_i \in \mathbb{R}^d$, forming a sequence $(G_{\mathbf{w},\lambda}\mathbf{x}_i)_{i=1}^n \in (\mathbb{R}^d)^n$. The standard transformer $\mathcal{T} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ is a well-defined function. Thus, the composition

$$\mathcal{GT}_{\mathbf{w},\lambda} = \mathcal{T} \circ \phi_{\mathbf{w},\lambda}$$

maps $X \in (\mathbb{R}^d)^n$ to a unique $Y \in (\mathbb{R}^d)^n$, ensuring well-definedness. □

Remark 4.2. *The Graded Transformer $\mathcal{GT}_{\mathbf{w},\lambda}$ is generally non-linear, as the standard transformer $\mathcal{T}$ includes non-linear operations, such as the sf function and feed-forward layers with activation functions (e.g., ReLU). However, $\phi_{\mathbf{w},\lambda}$ is linear, since $G_{\mathbf{w},\lambda}$ is a linear transformation applied component-wise.*

The **Graded Attention** operator modifies self-attention to prioritize features according to $\mathbf{w}$, defined as

$$\mathbf{A}_{\mathbf{w},\lambda}(Q, K, V) = \mathrm{sf}\left(\frac{QG_{\mathbf{w},\lambda}K^T}{\sqrt{d_k}}\right)V,$$

where $Q, K, V \in \mathrm{Mat}_{n,d_k}(\mathbb{R})$.

Graded attention employs a positive definite bilinear form on $\mathbb{R}^{d_k}$, defined by

$$\langle \mathbf{q}_i, \mathbf{k}_j \rangle_{\mathbf{w},\lambda} = \mathbf{q}_i^T G_{\mathbf{w},\lambda}\mathbf{k}_j = \sum_{k=0}^{d_k-1} \lambda^{q_k} q_{ik}k_{jk},$$

weighting similarities by grades, with positive definiteness shown below.

Lemma 4.3. *For $\lambda > 1$, $q_i \geq 0$, the graded attention weights*

$$\alpha_{ij} = \mathrm{sf}\left(\frac{\mathbf{q}_i^T G_{\mathbf{w},\lambda}\mathbf{k}_j}{\sqrt{d_k}}\right)_{ij}$$

concentrate on features with grades close to $q_{\max}$, with decay rate $O(\lambda^{q_k - q_{\max}})$ for $q_k < q_{\max}$, as $\lambda \to \infty$.

Proof. The attention score $s_{ij} = \mathbf{q}_i^T G_{\mathbf{w},\lambda}\mathbf{k}_j = \sum_{k=0}^{d_k-1} \lambda^{q_k} q_{ik}k_{jk}$. For $\lambda > 1$, terms with $q_k \approx q_{\max}$ dominate, as

$$\frac{\lambda^{q_k}}{\lambda^{q_{\max}}} = \lambda^{q_k - q_{\max}}$$

decays exponentially. Assuming $|q_{ik}|, |k_{jk}| \leq C$, the score approximates

$$s_{ij} \approx \lambda^{q_{\max}} \sum_{k:q_k = q_{\max}} q_{ik}k_{jk},$$

with error $O(\lambda^{q_k - q_{\max}})$. The sf function

$$\alpha_{ij} \propto \exp(s_{ij}/\sqrt{d_k})$$

concentrates on indices j with high-grade features, with subdominant terms decaying at $O(\lambda^{q_k - q_{\max}})$. □

Proposition 4.4. *The bilinear form $\langle \cdot, \cdot \rangle_{\mathbf{w},\lambda}$ on $\mathbb{R}^{d_k}$ is positive definite for $\lambda > 0$.*

Proof. For $\mathbf{x} \in \mathbb{R}^{d_k}$,

$$\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbf{w},\lambda} = \mathbf{x}^T G_{\mathbf{w},\lambda} \mathbf{x} = \sum_{i=0}^{d_k - 1} \lambda^{q_i} x_i^2.$$

Since $\lambda > 0$, $\lambda^{q_i} > 0$. If $\mathbf{x} \neq 0$, some $x_i \neq 0$, so the sum is positive; if $\mathbf{x} = 0$, the sum is zero. □

Proposition 4.5. *The attention score $s_{ij} = \mathbf{q}_i^T G_{\mathbf{w},\lambda} \mathbf{k}_j$ is Lipschitz continuous with respect to $\mathbf{q}_i, \mathbf{k}_j$, with constant at most $\lambda^{q_{\max}} C$, where $C > 0$ bounds $\|\mathbf{q}_i\|_2, \|\mathbf{k}_j\|_2$.*

Proof. For $\mathbf{q}_i, \mathbf{q}_i', \mathbf{k}_j, \mathbf{k}_j'$,

$$\begin{aligned} |s_{ij} - s_{ij}'| &= |(\mathbf{q}_i - \mathbf{q}_i')^T G_{\mathbf{w},\lambda} \mathbf{k}_j + \mathbf{q}_i'^T G_{\mathbf{w},\lambda} (\mathbf{k}_j - \mathbf{k}_j')| \\ &\leq \|\mathbf{q}_i - \mathbf{q}_i'\|_2 \|G_{\mathbf{w},\lambda} \mathbf{k}_j\|_2 + \|\mathbf{q}_i'\|_2 \|G_{\mathbf{w},\lambda} (\mathbf{k}_j - \mathbf{k}_j')\|_2. \end{aligned}$$

By Lem. 2.3, $\|G_{\mathbf{w},\lambda}\|_2 = \lambda^{q_{\max}}$, so

$$\|G_{\mathbf{w},\lambda} \mathbf{k}_j\|_2 \leq \lambda^{q_{\max}} \|\mathbf{k}_j\|_2, \quad \|G_{\mathbf{w},\lambda} (\mathbf{k}_j - \mathbf{k}_j')\|_2 \leq \lambda^{q_{\max}} \|\mathbf{k}_j - \mathbf{k}_j'\|_2.$$

With $\|\mathbf{q}_i'\|_2, \|\mathbf{k}_j\|_2 \leq C$,

$$|s_{ij} - s_{ij}'| \leq \lambda^{q_{\max}} C (\|\mathbf{q}_i - \mathbf{q}_i'\|_2 + \|\mathbf{k}_j - \mathbf{k}_j'\|_2).$$

□

Theorem 4.6. *Let $\mathcal{T}_{\mathbf{w},\lambda} = \mathcal{T} \circ \phi_{\mathbf{w},\lambda}$ be a Graded Transformer, where $\mathcal{T}$ is a standard transformer and $\phi_{\mathbf{w},\lambda}$ is the componentwise grading transformation. Then $\mathcal{T}_{\mathbf{w},\lambda}$ is a universal approximator for continuous sequence-to-sequence functions on compact domains. Moreover, for target functions that exhibit hierarchical structure aligned with the grading $\mathbf{w}$, the Graded Transformer may achieve comparable approximation accuracy with fewer parameters than a standard transformer.*

Proof. By [36], standard transformers $\mathcal{T}$ are universal approximators for sequence-to-sequence mappings over compact domains. The grading transformation $\phi_{\mathbf{w},\lambda}$, defined by $x_i \mapsto G_{\mathbf{w},\lambda} x_i$, is an invertible, smooth, and Lipschitz-continuous map (Lem. 2.3). Hence, the composition $\mathcal{T}_{\mathbf{w},\lambda} = \mathcal{T} \circ \phi_{\mathbf{w},\lambda}$ retains the universal approximation property by function composition stability.

For functions f that are themselves invariant under certain hierarchical scalings — for example, functions where only high-grade features dominate the output — the composition $\mathcal{T}_{\mathbf{w},\lambda}$ can focus modeling capacity on fewer effective degrees of freedom. This may reduce the number of parameters required to approximate f to a given accuracy. However, this claim is qualitative and depends on alignment between f and the grading $\mathbf{w}$; it is not a general guarantee. □

Proposition 4.7. *Let $\lambda > 1$ and $q_i \geq 0$. Let $\mathbf{A}_{\mathbf{w},\lambda}(Q,K,V)$ denote the graded attention output and $\mathbf{A}(Q,K,V)$ the standard attention. Then the singular values of $QG_{\mathbf{w},\lambda}K^T$ are scaled by at most $\lambda^{q_{\max}}$ compared to those of QK^T, where $q_{\max} = \max_i q_i$. Consequently, the effective numerical rank of $\mathbf{A}_{\mathbf{w},\lambda}(Q,K,V)$ may increase or decrease depending on the spectral distribution of QK^T and the grading tuple $\mathbf{w}$.*

Proof. Let $S = QG_{\mathbf{w},\lambda}K^T/\sqrt{d_k}$, and suppose $QK^T = U\Sigma V^T$ is the singular value decomposition with singular values $\sigma_i > 0$. Since $G_{\mathbf{w},\lambda}$ is diagonal with $\|G_{\mathbf{w},\lambda}\|_2 = \lambda^{q_{\max}}$, we have:

$$\|QG_{\mathbf{w},\lambda}K^T\|_2 \leq \|Q\|_2 \cdot \|G_{\mathbf{w},\lambda}\|_2 \cdot \|K^T\|_2 \leq \lambda^{q_{\max}}\|QK^T\|_2.$$

Each singular value $\tilde{\sigma}_i$ of $QG_{\mathbf{w},\lambda}K^T$ is therefore bounded above by $\lambda^{q_{\max}}\sigma_i$. Whether this amplification increases the number of singular values above a numerical threshold (e.g., ϵ) depends on how the grading affects components with lower weights. In particular, if some $q_i \ll q_{\max}$, the corresponding components may be suppressed, potentially decreasing effective rank. Thus, rank enhancement is possible but not guaranteed. □

Proposition 4.8. *For any matrix $A_0 \in \mathrm{Mat}_n(\mathbb{R})$ with non-negative entries, there exist $Q, K, \mathbf{w}, \lambda$ such that $\mathbf{A}_{\mathbf{w},\lambda}(Q,K,V)$ approximates A_0 with*

$$\|A - A_0\|_F \leq \delta.$$

Proof. For A_0 with $a_{ij} \geq 0$, let A_0' replace zeros with $\epsilon = \delta/(2\sqrt{n})$. Choose $Q, K \in \mathrm{Mat}_{n,d_k}(\mathbb{R})$ such that

$$QK^T = \sqrt{d_k}\log A_0'.$$

Set $\mathbf{w}$ with $q_i = ci$, $c > 0$, and large λ. By Lem. 4.3, $\mathbf{A}_{\mathbf{w},\lambda}$ approximates A_0' with $\|A - A_0'\|_F \leq \delta/2$. Since $\|A_0' - A_0\|_F \leq \delta/2$, the error is $\|A - A_0\|_F \leq \delta$. □

Proposition 4.9. *For functions $\mathcal{F}$ with weights $w_i \propto e^{q_i}$, $q_i \geq 0$, the Graded Transformer's VC dimension is reduced by*

$$\sum_{i=0}^{d-1} e^{-q_i},$$

lowering sample complexity.

Proof. For $\mathcal{F}$, $G_{\mathbf{w},\lambda}$ aligns the hypothesis space with $\mathbf{w}$. The VC dimension of $\mathcal{T} \circ \phi_{\mathbf{w},\lambda}$ is reduced by

$$\sum_{i=0}^{d-1} e^{-q_i},$$

lowering sample complexity [15]. □

Proposition 4.10. *For noise Δ, $\|\Delta\|_2 \leq \epsilon$, the error is*

$$\|\mathcal{GT}_{\mathbf{w},\lambda}(X+\Delta) - \mathcal{GT}_{\mathbf{w},\lambda}(X)\|_2 \leq L\lambda^{q_{\max}}\epsilon,$$

where L is the Lipschitz constant of $\mathcal{GT}_{\mathbf{w},\lambda}$.

Proof.

$$\|\mathcal{GT}_{\mathbf{w},\lambda}(X+\Delta) - \mathcal{GT}_{\mathbf{w},\lambda}(X)\|_2 \leq L_{\mathcal{T}}\|\phi_{\mathbf{w},\lambda}(X+\Delta) - \phi_{\mathbf{w},\lambda}(X)\|_2.$$

For $\phi_{\mathbf{w},\lambda}(X) = (G_{\mathbf{w},\lambda}\mathbf{x}_i)$, Lem. 2.5 gives

$$\|\phi_{\mathbf{w},\lambda}(X+\Delta) - \phi_{\mathbf{w},\lambda}(X)\|_2 \leq \lambda^{q_{\max}}\epsilon.$$

By Prop. 2.6, $L = \prod_l L_{\sigma_l}\|W_l\|_2\lambda^{q_{\max}}$, so the error is

$$L\lambda^{q_{\max}}\epsilon.$$

□

Proposition 4.11. *The graded attention $\mathbf{A}_{\mathbf{w},\lambda}(Q,K,V)$ has complexity*

$$O(n^2 d_k + nd_k^2),$$

with an additional $O(nd_k)$ cost for grading.

Proof. Computing $QG_{\mathbf{w},\lambda}K^T$ costs $O(nd_k)$ for $G_{\mathbf{w},\lambda}$, plus $O(n^2 d_k)$ for the product. The sf function and multiplication by V cost $O(n^2 d_k + nd_k^2)$, matching standard attention, with grading adding $O(nd_k)$. □

Remark 4.12. *The Graded Transformer's properties, including attention concentration (Lem. 4.3) and expressivity (Prop. 4.8), leverage grading to model hierarchical data efficiently. The assumptions $q_i \geq 0$, $\lambda > 1$ ensure positive scaling, but arbitrary q_i could be considered with adjusted bounds ($\max_i |\lambda^{q_i}|$).*

5. Architecture of Graded Transformers

This section details the architectural components of the Graded Transformer, building on the framework established in Sections 2 and 4. By integrating grading transformations $G_{\mathbf{w},\lambda}$ across inputs, positional encodings, attention mechanisms, feed-forward layers, and output layers, the Graded Transformer embeds hierarchical priors to enhance feature prioritization and efficiency for structured sequence data [15, 21]. Each component is designed to amplify high-grade features, aligning with the transformer's dynamic learning capabilities [32]. We provide rigorous mathematical formulations, motivate the design choices, and prove all stability and expressivity properties, extending the foundational work on graded neural architectures [15].

5.1. Graded Input Representation.

The input representation transforms raw token embeddings to emphasize features based on their grades, ensuring that hierarchical structures are captured from the outset. For each token $\mathbf{x}_i \in \mathbb{R}^d$ in the input sequence $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, we apply the grading transformation from Section 2:

$$\mathbf{x}_i' = G_{\mathbf{w},\lambda}\mathbf{x}_i, \quad G_{\mathbf{w},\lambda} = \operatorname{diag}(\lambda^{q_0}, \ldots, \lambda^{q_{d-1}}),$$

where $\mathbf{w} = (q_0, \ldots, q_{d-1})$, $q_i \in \mathbb{R}$, and $\lambda > 0$. To prevent numerical instability due to large λ^{q_i}, we normalize:

$$\mathbf{x}_i'' = \frac{\mathbf{x}_i'}{\|\mathbf{x}_i'\|_2}.$$

The graded and normalized token is then processed through a linear layer with activation:

$$\mathbf{h}_i = \sigma(W\mathbf{x}_i'' + \mathbf{b}),$$

where $W \in \mathbb{R}^{m\times d}$, $\mathbf{b} \in \mathbb{R}^m$, and σ is typically ReLU.

Theorem 5.1 (Input Stability)**.** *The mapping $\mathbf{x}_i \mapsto \mathbf{x}_i''$ is Lipschitz continuous with constant at most $\lambda^{q_{\max}}$, where $q_{\max} = \max_{i=0,\ldots,d-1} q_i$, assuming $\lambda > 1$ and $q_i \geq 0$.*

Proof. Consider the mapping $\mathbf{x}_i \mapsto \mathbf{x}_i'' = \frac{G_{\mathbf{w},\lambda}\mathbf{x}_i}{\|G_{\mathbf{w},\lambda}\mathbf{x}_i\|_2}$. For inputs $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, first analyze the grading step:

$$\mathbf{x}' = G_{\mathbf{w},\lambda}\mathbf{x}, \quad \mathbf{y}' = G_{\mathbf{w},\lambda}(\mathbf{y}).$$

From Section 2, $\|G_{\mathbf{w},\lambda}(\mathbf{x}-\mathbf{y})\|_2 \leq \lambda^{q_{\max}}\|\mathbf{x}-\mathbf{y}\|_2$. Thus:

$$\|\mathbf{x}'-\mathbf{y}'\|_2 = \|G_{\mathbf{w},\lambda}(\mathbf{x}-\mathbf{y})\|_2 \leq \lambda^{q_{\max}}\|\mathbf{x}-\mathbf{y}\|_2.$$

Next, the normalization step $\mathbf{z} \mapsto \frac{\mathbf{z}}{\|\mathbf{z}\|_2}$ is 1-Lipschitz for $\mathbf{z} \neq 0$. Let $\mathbf{x}'' = \frac{\mathbf{x}'}{\|\mathbf{x}'\|_2}$, $\mathbf{y}'' = \frac{\mathbf{y}'}{\|\mathbf{y}'\|_2}$. The distance is:

$$\|\mathbf{x}''-\mathbf{y}''\|_2 = \left\| \frac{\mathbf{x}'}{\|\mathbf{x}'\|_2} - \frac{\mathbf{y}'}{\|\mathbf{y}'\|_2} \right\|_2 \leq \frac{\|\mathbf{x}'-\mathbf{y}'\|_2}{\min(\|\mathbf{x}'\|_2, \|\mathbf{y}'\|_2)}.$$

Assuming $\|\mathbf{x}'\|_2, \|\mathbf{y}'\|_2 \geq \epsilon > 0$ (ensured by non-zero inputs and regularization in practice), we have:

$$\|\mathbf{x}''-\mathbf{y}''\|_2 \leq \frac{\lambda^{q_{\max}}\|\mathbf{x}-\mathbf{y}\|_2}{\epsilon}.$$

For simplicity, if inputs are normalized ($\|\mathbf{x}\|_2, \|\mathbf{y}\|_2 \approx 1$), the constant is approximately $\lambda^{q_{\max}}$. □

Corollary 5.2 (Bounded Activations)**.** *For all i, $\|\mathbf{x}_i''\|_2 = 1$.*

Proof. By definition, $\mathbf{x}_i'' = \frac{\mathbf{x}_i'}{\|\mathbf{x}_i'\|_2}$, so:

$$\|\mathbf{x}_i''\|_2 = \left\| \frac{\mathbf{x}_i'}{\|\mathbf{x}_i'\|_2} \right\|_2 = 1,$$

assuming $\mathbf{x}_i' \neq 0$. □

Lemma 5.3 (Jacobian Bound)**.** *The Jacobian of the mapping $\mathbf{x}_i \mapsto \mathbf{x}_i'$ has operator norm $\lambda^{q_{\max}}$.*

Proof. The mapping is $\mathbf{x}_i \mapsto \mathbf{x}_i' = G_{\mathbf{w},\lambda}\mathbf{x}_i$. Since $G_{\mathbf{w},\lambda}$ is linear, the Jacobian is $G_{\mathbf{w},\lambda}$, a diagonal matrix with entries λ^{q_i}. The operator norm is the maximum singular value, which for a diagonal matrix is:

$$\|G_{\mathbf{w},\lambda}\|_2 = \max_i \lambda^{q_i} = \lambda^{q_{\max}}.$$

□

5.2. **Graded Positional Encoding.** Positional encodings are critical for transformers to capture sequence order. We enhance them with grading transformations to prioritize certain positions, such as earlier tokens in hierarchical tasks like parsing. Standard positional encodings are:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right),$$

for position pos and dimension i. We grade them as:

$$PE'(pos, i) = \lambda^{w_{pos}} PE(pos, i), \quad w_{pos} = f(pos),$$

where $f(pos)$ is a grading function, typically $f(pos) = -\alpha pos$, $\alpha > 0$, to emphasize earlier positions. The input to the attention mechanism is:

$$\mathbf{z}_i = \mathbf{x}_i'' + PE'(pos_i, \cdot), \quad \mathbf{z}_i' = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2},$$

and attention scores are computed as:

$$(QK^T)_{ij} = (\mathbf{z}'_i W_Q)(\mathbf{z}'_j W_K)^T,$$

where $W_Q, W_K \in \mathbb{R}^{d \times d_k}$.

Proposition 5.4 (Positional Bias)**.** *For $f(pos) = -\alpha pos$, $\alpha > 0$, the graded attention biases earlier positions.*

Proof. The graded encoding is:

$$PE'(pos, i) = \lambda^{-\alpha pos} PE(pos, i).$$

For $\lambda > 1$, $\lambda^{-\alpha pos} = (\lambda^\alpha)^{-pos}$ decreases as pos increases, so earlier positions (pos small) have larger scaling factors. In the attention score:

$$\langle \mathbf{z}'_i, \mathbf{z}'_j \rangle_{\mathbf{w},\lambda} \approx \langle \mathbf{x}''_i + \lambda^{-\alpha pos_i} PE(pos_i), \mathbf{x}''_j + \lambda^{-\alpha pos_j} PE(pos_j) \rangle_{\mathbf{w},\lambda},$$

earlier positions contribute larger terms due to higher $\lambda^{-\alpha pos}$, biasing attention toward them. □

Lemma 5.5 (Positional Stability)**.** *The mapping $pos \mapsto \mathbf{z}'_i$ is Lipschitz continuous with constant bounded by $C\lambda^{|w_{\max}|}$, where $w_{\max} = \max_{pos} |f(pos)|$.*

Proof. Consider $\mathbf{z}_i = \mathbf{x}''_i + PE'(pos, \cdot)$, $\mathbf{z}'_i = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2}$. For positions pos, pos':

$$\|\mathbf{z}_i(pos) - \mathbf{z}_i(pos')\|_2 = \|PE'(pos, \cdot) - PE'(pos', \cdot)\|_2.$$

Since $PE(pos, i)$ is bounded ($|PE(pos, i)| \leq 1$), we have:

$$\|PE'(pos, i)\|_2 \leq \lambda^{f(pos)}, \quad \|PE'(pos, i) - PE'(pos', i)\|_2 \leq |\lambda^{f(pos)} - \lambda^{f(pos')}| \cdot |PE(pos, i)|.$$

For $f(pos) = -\alpha pos$, assume $\lambda > 1$. The difference is:

$$|\lambda^{-\alpha pos} - \lambda^{-\alpha pos'}| = \lambda^{-\alpha \min(pos, pos')} |\lambda^{\alpha |pos - pos'|} - 1|.$$

For small $|pos - pos'|$, use the mean value theorem:

$$|\lambda^{\alpha t} - 1| \leq \alpha \ln \lambda \cdot \lambda^{\alpha t} \cdot |t|,$$

so:

$$\|PE'(pos, \cdot) - PE'(pos', \cdot)\|_2 \leq C\lambda^{|f(pos)|} |pos - pos'|,$$

where C depends on α, $\ln \lambda$, and the dimension d. Normalization is 1-Lipschitz, so the constant is bounded by $C\lambda^{|w_{\max}|}$. □

5.3. **Graded Attention Mechanism.** The attention mechanism is the core of transformers, capturing dependencies between tokens. We introduce grading transformations to prioritize high-grade features in attention scores, enhancing the model's focus on hierarchically significant tokens. The base attention is:

$$\text{Attention}(Q, K, V) = \text{sf}\left(\frac{QK^T}{\sqrt{d_k}}\right) V.$$

We propose four graded attention variants, each applying $G_{\mathbf{w},\lambda}$ differently:

(1) **Graded Scores**:

$$\text{Score}_{ij} = \sum_{k=1}^{d_k} \lambda^{w_k} q_{ik} k_{jk}, \quad G^K_{\mathbf{w},\lambda} = \text{diag}(\lambda^{w_1}, \ldots, \lambda^{w_{d_k}}),$$

$$\text{Score}_{ij} = (QG^K_{\mathbf{w},\lambda} K^T)_{ij}.$$

This weights each dimension's contribution by its grade.

(2) **Graded Queries/Keys**:

$$Q' = G_{\mathbf{w},\lambda} Q, \quad K' = G_{\mathbf{w},\lambda} K, \quad \text{Score}_{ij} = \langle \mathbf{q}_i, \mathbf{k}_j \rangle_{\mathbf{w},\lambda} = \mathbf{q}_i^T G_{\mathbf{w},\lambda} \mathbf{k}_j.$$

This scales queries and keys before computing scores.

(3) **Graded Multi-Head**:

$$\text{Head}_h = \text{Attention}(G_{\mathbf{w}_h,\lambda} Q_h, G_{\mathbf{w}_h,\lambda} K_h, V_h),$$

with distinct $\mathbf{w}_h$ per head, allowing head-specific grading.

(4) **Graded Values**:

$$V' = G_{\mathbf{w},\lambda} V, \quad \mathbf{o}_i = \sum_{j=1}^{n} \alpha_{ij} (G_{\mathbf{w},\lambda} \mathbf{v}_j),$$

$$\alpha_{ij} = \text{sf}\left(\frac{\mathbf{q}_i^T \mathbf{k}_j}{\sqrt{d_k}} \right).$$

This scales the output values, emphasizing high-grade features.

Theorem 5.6 (Attention Stability)**.** *For the Graded Queries/Keys variant, the score $\text{Score}_{ij} = \mathbf{q}_i^T G_{\mathbf{w},\lambda} \mathbf{k}_j$ is Lipschitz continuous with constant at most $\lambda^{q_{\max}} C$.*

Proof. For $\mathbf{q}_i, \mathbf{q}_i', \mathbf{k}_j, \mathbf{k}_j'$:

$$|\mathbf{q}_i^T G_{\mathbf{w},\lambda} \mathbf{k}_j - \mathbf{q}_i'^T G_{\mathbf{w},\lambda} \mathbf{k}_j'| \leq \|\mathbf{q}_i - \mathbf{q}_i'\|_2 \|G_{\mathbf{w},\lambda} \mathbf{k}_j\|_2 + \|\mathbf{q}_i'\|_2 \|G_{\mathbf{w},\lambda} (\mathbf{k}_j - \mathbf{k}_j')\|_2.$$

From Section 2:

$$\|G_{\mathbf{w},\lambda} \mathbf{k}_j\|_2 \leq \lambda^{q_{\max}} \|\mathbf{k}_j\|_2, \quad \|G_{\mathbf{w},\lambda} (\mathbf{k}_j - \mathbf{k}_j')\|_2 \leq \lambda^{q_{\max}} \|\mathbf{k}_j - \mathbf{k}_j'\|_2.$$

With $\|\mathbf{k}_j\|_2, \|\mathbf{q}_i'\|_2 \leq C$:

$$|\mathbf{q}_i^T G_{\mathbf{w},\lambda} \mathbf{k}_j - \mathbf{q}_i'^T G_{\mathbf{w},\lambda} \mathbf{k}_j'| \leq \lambda^{q_{\max}} C (\|\mathbf{q}_i - \mathbf{q}_i'\|_2 + \|\mathbf{k}_j - \mathbf{k}_j'\|_2).$$

□

Proposition 5.7 (Head Diversity)**.** *Distinct grading tuples $\mathbf{w}_h$ in the Graded Multi-Head variant enhance representational capacity.*

Proof. In the Graded Multi-Head variant, each head computes:

$$\text{Head}_h = \text{sf}\left(\frac{(G_{\mathbf{w}_h,\lambda} Q_h)(G_{\mathbf{w}_h,\lambda} K_h)^T}{\sqrt{d_k}} \right) V_h.$$

Distinct $\mathbf{w}_h = (q_{h,0}, \ldots, q_{h,d_k-1})$ produce unique $G_{\mathbf{w}_h,\lambda}$, scaling query and key dimensions differently. This projects each head onto a distinct graded subspace, as the singular values of $G_{\mathbf{w}_h,\lambda} Q_h K_h^T G_{\mathbf{w}_h,\lambda}^T$ vary with $\mathbf{w}_h$. The concatenated heads span a richer subspace of $\mathbb{R}^d$, enhancing the model's ability to capture diverse dependencies compared to uniform grading. □

5.4. **Graded Feed-Forward Layers.** Feed-forward layers process token representations independently, and grading ensures that outputs reflect hierarchical priorities. The standard feed-forward network (FFN) is:

$$\mathrm{FFN}(\mathbf{x}) = \mathrm{ReLU}(\mathbf{x}W_1 + \mathbf{b}_1)W_2 + \mathbf{b}_2,$$

where $W_1 \in \mathbb{R}^{d \times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d}$, and d_{ff} is the hidden dimension. The graded FFN is:

$$\mathrm{FFN}'(\mathbf{x}) = G_{\mathbf{w},\lambda}\mathrm{FFN}(\mathbf{x}), \quad \mathbf{h}' = \frac{\mathrm{FFN}'(\mathbf{x})}{\|\mathrm{FFN}'(\mathbf{x})\|_2},$$

applying grading to prioritize features, followed by normalization for stability.

Proposition 5.8 (FFN Stability)**.** *The graded FFN mapping $\mathbf{x} \mapsto FFN'(\mathbf{x})$ has Lipschitz constant at most $\lambda^{q_{\max}} L_{FFN}$.*

Proof. Let $\mathrm{FFN}'(\mathbf{x}) = G_{\mathbf{w},\lambda}\mathrm{FFN}(\mathbf{x})$. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$:

$$\|\mathrm{FFN}'(\mathbf{x}) - \mathrm{FFN}'(\mathbf{y})\|_2 = \|G_{\mathbf{w},\lambda}(\mathrm{FFN}(\mathbf{x}) - \mathrm{FFN}(\mathbf{y}))\|_2 \leq \lambda^{q_{\max}} \|\mathrm{FFN}(\mathbf{x}) - \mathrm{FFN}(\mathbf{y})\|_2,$$

using the norm bound from Section 2. Since FFN is Lipschitz with constant L_{FFN} (due to ReLU and linear layers):

$$\|\mathrm{FFN}(\mathbf{x}) - \mathrm{FFN}(\mathbf{y})\|_2 \leq L_{\mathrm{FFN}} \|\mathbf{x} - \mathbf{y}\|_2.$$

Thus:

$$\|\mathrm{FFN}'(\mathbf{x}) - \mathrm{FFN}'(\mathbf{y})\|_2 \leq \lambda^{q_{\max}} L_{\mathrm{FFN}} \|\mathbf{x} - \mathbf{y}\|_2.$$

Normalization is 1-Lipschitz, so the constant remains $\lambda^{q_{\max}} L_{\mathrm{FFN}}$. □

5.5. **Graded Output Layer.** The output layer produces the final predictions, and grading ensures that hierarchical priorities are reflected in the results. The standard output is a linear layer followed by softmax:

$$\mathbf{z} = W_{\mathrm{out}}\mathbf{h} + \mathbf{b}_{\mathrm{out}}, \quad \mathrm{sf}(\mathbf{z}).$$

The graded output is:

$$\mathbf{h}' = G_{\mathbf{w},\lambda}\mathbf{h}, \quad \mathbf{z} = W_{\mathrm{out}}\mathbf{h}' + \mathbf{b}_{\mathrm{out}},$$

emphasizing high-grade features before the final projection.

Proposition 5.9 (Output Stability)**.** *The output mapping $\mathbf{h} \mapsto \mathbf{z}$ is Lipschitz with constant at most $\lambda^{q_{\max}} L_{out}$.*

Proof. For $\mathbf{h}_1, \mathbf{h}_2 \in \mathbb{R}^d$:

$$\mathbf{z}_1 = W_{\mathrm{out}}(G_{\mathbf{w},\lambda}\mathbf{h}_1) + \mathbf{b}_{\mathrm{out}}, \quad \mathbf{z}_2 = W_{\mathrm{out}}(G_{\mathbf{w},\lambda}\mathbf{h}_2) + \mathbf{b}_{\mathrm{out}}.$$

Thus:

$$\|\mathbf{z}_1 - \mathbf{z}_2\|_2 = \|W_{\mathrm{out}}G_{\mathbf{w},\lambda}(\mathbf{h}_1 - \mathbf{h}_2)\|_2 \leq \|W_{\mathrm{out}}\|_2 \|G_{\mathbf{w},\lambda}(\mathbf{h}_1 - \mathbf{h}_2)\|_2.$$

From Section 2:

$$\|G_{\mathbf{w},\lambda}(\mathbf{h}_1 - \mathbf{h}_2)\|_2 \leq \lambda^{q_{\max}} \|\mathbf{h}_1 - \mathbf{h}_2\|_2.$$

Let $L_{\mathrm{out}} = \|W_{\mathrm{out}}\|_2$. Then:

$$\|\mathbf{z}_1 - \mathbf{z}_2\|_2 \leq \lambda^{q_{\max}} L_{\mathrm{out}} \|\mathbf{h}_1 - \mathbf{h}_2\|_2.$$

□

Proposition 5.10 (Computational Complexity)**.** *The Graded Transformer has the same asymptotic complexity as the standard transformer, $O(n^2 d + nd^2)$, with additional $O(nd)$ cost for grading transformations.*

Proof. The standard transformer's complexity is dominated by attention ($O(n^2d)$) and feed-forward layers ($O(nd^2)$). Each grading transformation $G_{\mathbf{w},\lambda}$ is a diagonal matrix multiplication, costing $O(d)$ per token, or $O(nd)$ for n tokens. This is applied to inputs, encodings, attention, feed-forward, and output layers, adding $O(nd)$ per component. Since $nd \ll n^2d, nd^2$, the overall complexity remains $O(n^2d+nd^2)$. □

Remark 5.11. *The architecture of the Graded Transformer systematically integrates grading transformations to prioritize hierarchical features, extending the GNN framework* [21] *to sequence modeling. By applying grading across all components, it ensures consistent feature emphasis, improving efficiency for structured data. The stability properties guarantee robustness, while the computational complexity remains comparable to standard transformers. Future work includes optimizing* $\mathbf{w}$ *and* λ*, potentially via gradient descent, and empirically validating performance on tasks like syntactic parsing or physical system modeling* [15, 21].

6. Training and Optimization

Training the Graded Transformer involves optimizing its parameters to balance hierarchical feature prioritization with predictive accuracy. This process leverages the mathematical properties developed in Section 4, including Lipschitz continuity, attention amplification, and stability. This section defines the graded loss function, introduces regularization, describes optimization strategies, and states convergence guarantees relevant to gradient-based learning. Applications include structured domains such as algebraic geometry and natural language processing [15, 21].

6.1. Graded Loss Function.

To align the learning objective with the model's hierarchical inductive bias, we define a grade-weighted loss:

$$\mathcal{L} = \sum_{i=1}^{m} \sum_{k=1}^{d} \lambda^{q_k} \, \ell(\hat{y}_{i,k}, y_{i,k}),$$

where:

- $\ell(\hat{y}_{i,k}, y_{i,k})$ is a base loss function (e.g., cross-entropy),
- $\hat{y}_{i,k}, y_{i,k} \in \mathbb{R}$ are predicted and true outputs for the k-th output dimension of token i,
- m is the sequence length and d is the output dimension,
- λ^{q_k} emphasizes loss in high-grade components, reflecting their hierarchical importance.

6.2. Regularization and Optimization.

If the grade vector $\mathbf{w} = (q_1, \ldots, q_d)$ is learned, we add a regularization term to penalize excessively large grades:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \gamma \|\mathbf{w}\|_2^2,$$

with regularization weight $\gamma > 0$. Optimization proceeds using standard gradient-based methods such as Adam. Differentiation through the exponential grading map introduces a sensitivity factor. For example, the gradient of a single attention score with respect to grade q_k is:

$$\frac{\partial \text{Score}_{ij}}{\partial q_k} = \lambda^{q_k} \ln \lambda \cdot q_{ik} k_{jk},$$

where q_{ik}, k_{jk} denote the k-th components of the query and key vectors at positions i and j, respectively.

6.3. **Convergence and Gradient Stability.**

Theorem 6.1 (Convergence)**.** *Fix a grading vector* $\mathbf{w}$*. If the loss function* ℓ *is Lipschitz continuous, then gradient descent with sufficiently small step size converges to a stationary point of the Graded Transformer's loss.*

Proof. The Graded Transformer $\mathcal{GT}_{\mathbf{w},\lambda}$ is Lipschitz continuous by **??**, with Lipschitz constant scaling as $L_{\mathcal{T}}\lambda^{q_{\max}}$. Composing with a Lipschitz loss function yields a Lipschitz objective. Convergence to a stationary point follows from classical results on gradient descent for smooth functions, provided the step size η satisfies $\eta < 2/L$ for the global Lipschitz constant L of the gradient. □

Proposition 6.2 (Gradient Stability)**.** *Assume that* $\|\mathbf{q}_i\|_2, \|\mathbf{k}_j\|_2 \leq C$ *and that* $\left|\frac{\partial \ell}{\partial \hat{y}_{i,j}}\right| \leq L_\ell$*. Then the gradient* $\frac{\partial \mathcal{L}_{\text{total}}}{\partial q_k}$ *is Lipschitz continuous in* q_k*, with constant proportional to* $\lambda^{q_{\max}} \ln \lambda$*.*

Proof. The form of the derivative is given by:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial q_k} = \sum_{i=1}^{m}\sum_{j=1}^{d}\left(\lambda^{q_j}\frac{\partial \ell}{\partial \hat{y}_{i,j}} \cdot \frac{\partial \hat{y}_{i,j}}{\partial q_k} + \ell(\hat{y}_{i,j}, y_{i,j}) \ln \lambda \cdot \lambda^{q_k}\delta_{j,k}\right) + 2\gamma q_k.$$

The terms involving λ^{q_j} and $\lambda^{q_k} \ln \lambda$ dominate the sensitivity. Under the boundedness assumptions, each term is Lipschitz in q_k, yielding an overall bound proportional to $\lambda^{q_{\max}} \ln \lambda$. □

Training the Graded Transformer leverages its hierarchical structure to bias learning toward semantically important features. While the architecture supports stable optimization in theory, optimizing $\mathbf{w}$ and tuning λ in practice may be challenging. Empirical techniques such as warm-starting, gradient clipping, and grade annealing may help. Future work should investigate performance on structured tasks such as equation parsing, tree-based inference, and symbolic regression [4, 11, 15–31].

7. Potential Applications

The Graded Transformer embeds hierarchical priors into its architecture via the transformation $G_{\mathbf{w},\lambda}$, enabling principled learning in structured domains. This section outlines potential applications across algebraic geometry, physics, natural language processing, biological sequence modeling, and cross-domain settings. Each of these fields exhibits natural grading—through degree, scale, structure, or functional relevance—which the model can exploit to improve attention efficiency (Prop. 4.7), sample complexity (Prop. 4.9), and interpretability.

7.1. **Algebraic Geometry.**

Graded structures are central in algebraic geometry, from the decomposition of polynomial rings to the geometry of moduli spaces and weighted projective varieties. The Graded Transformer aligns naturally with such contexts by emphasizing basis elements according to their algebraic degree or weighted monomial structure.

Tasks in this area that benefit from grading-aware architectures include:

- Modeling moduli spaces of genus two curves through theta constants and isogenies, where graded invariants govern the geometry [?sh-93, ?sh-91, ?2024-03].

- Computing zeta functions over weighted projective hypersurfaces, where the monomial weights determine the graded structure [?sh-94, ?sh-87].
- Learning properties of diagonalizable hypersurfaces, which are compatible with grading-based decomposition [?sh-100].
- Prioritizing leading terms in point-counting problems or symbolic computations involving weighted systems [?sh-94].

By embedding degree-based importance into attention weights, the model enhances both learning efficiency and symbolic interpretability. Mapping discrete degrees to continuous grades in $\mathbf{w}$ remains a design challenge requiring domain expertise.

7.2. **Physics.** Physical systems often exhibit multiscale behavior—ranging from microscopic energy levels to macroscopic spatial dynamics. Grading provides a mechanism to emphasize dominant scales, aligning the model with physical intuition. This applies across quantum mechanics, fluid dynamics, cosmology, and condensed matter theory.

Graded Transformers offer tools for modeling problems such as:

- Spectral prediction in quantum systems, with grading based on energy eigenvalues to prioritize high-energy orbitals [21].
- Large-eddy simulations in turbulence, emphasizing low-wavenumber structures through inverse-frequency grading [6].
- Time-series analysis in cosmology, where longer-term patterns (e.g., expansion phases) are made more salient by temporal grading [35].
- Modeling phase transitions, with grading focused on states near critical temperature to capture thermodynamic sensitivity [7].

The model's robustness to noise and ability to represent multiscale attention hierarchies make it well-suited to experimental and simulation data. However, continuous spectra and system-specific symmetries may require learned or hybrid grading mechanisms.

7.3. **Natural Language Processing.** Language has a well-defined hierarchical structure, with certain words and syntactic roles carrying more semantic weight. Graded Transformers capture this structure explicitly by adjusting attention strength according to token importance or position in a parse tree. Combined with positional grading functions (e.g., $f(\text{pos}) = -\alpha \cdot \text{pos}$), the model can encode both structural and sequential hierarchies.

Key linguistic applications where grading enhances performance include:

- Syntactic parsing, where syntactic heads and function words receive higher attention priority [3].
- Semantic role labeling and translation, which benefit from emphasizing predicate–argument structure [33].
- Dialogue understanding and intent classification, where critical intent-bearing tokens are elevated in attention [8].
- Question answering, where graded relevance scores can focus attention on semantically aligned spans [38].

The ability to make attention interpretable and linguistically grounded is a major advantage. Still, adapting $\mathbf{w}$ to different syntactic conventions (e.g., head-initial

vs. head-final languages) requires either domain-informed initialization or training from structured corpora.

7.4. **Biological Sequence Analysis.** Biological sequences encode functionally significant elements—such as regulatory regions, coding exons, or conserved protein motifs—within much larger noisy contexts. The Graded Transformer's capacity to prioritize key subsequences aligns well with these needs, especially in genomics and proteomics.

Biological tasks that benefit from attention guided by functional relevance include:

- Gene structure prediction, with emphasis on coding regions and known regulatory elements [2].
- Variant effect prediction, assigning higher weight to disease-associated SNPs [9].
- Protein structure and function modeling, particularly active site residues and conserved domains [34].
- Metagenomic classification, where grading focuses on taxonomic marker regions for microbial inference [37].

These applications are typically data-constrained, so the sample efficiency of grading is critical. Managing long sequences and designing functionally informed grading schemes remain open problems, though domain annotations (e.g., from UniProt or ENCODE) can support supervised learning of $\mathbf{w}$.

7.5. **Cross-Domain Applications.** Because grading abstracts the notion of importance, it transfers across domains—from algebraic degree and physical scale to syntactic function and biological relevance. This opens the door to novel transfer learning strategies and unified frameworks for modeling structured data.

Emerging opportunities for cross-domain application of Graded Transformers include:

- Pretraining on symbolic tasks (e.g., Gröbner basis learning [?sh-96]) followed by fine-tuning on NLP or genomics datasets.
- Encoding data fabrics as 4D graded structures, supporting flow-aware and curvature-sensitive learning [20, ?sh-99].
- Developing cross-domain benchmarks to empirically evaluate how shared grading principles improve generalization [10].

These directions connect Graded Transformers with broader goals in neurosymbolic learning [?sh-86, ?sh-85], unifying discrete mathematical structure with continuous learning. Scalability, data heterogeneity, and grading transferability are the key challenges ahead.

8. Closing Remarks

Graded Transformers represent more than an architectural refinement—they signal a shift in how structure, hierarchy, and mathematical priors can be encoded directly into modern learning systems. By introducing grading transformations into the attention and representation layers of transformer models, we open the door to architectures that are not only data-driven, but also geometry-aware and algebraically grounded. This graded perspective offers a powerful new lens through which to design models for scientific, symbolic, and structured data domains.

The potential of Graded Transformers extends far beyond their initial formulation. In algebraic geometry, they invite a new generation of models capable of learning over moduli spaces, isogeny classes, or weighted projective hypersurfaces. In physics, they suggest architectures attuned to the natural hierarchies of scale, symmetry, and phase structure. In language, they provide a mechanism for embedding linguistic or syntactic roles into the model's inductive bias. In genomics, they offer a way to prioritize functional regions in vast, noisy biological sequences.

A number of open directions remain. Learning or discovering optimal grading schemes—whether through optimization, symbolic heuristics, or geometric constraints—is a central challenge. Extending graded architectures beyond transformers, to include graph networks, recurrent models, or hybrid neuro-symbolic pipelines, could reveal deeper structural synergies. Transfer learning across domains with shared grading principles (e.g., from algebraic systems to proteins or syntax) may redefine how we train generalizable, low-sample complexity models. The integration of grading with sparse attention, equivariant networks, or Finsler-geometric representations presents opportunities for both theoretical development and practical efficiency.

Ultimately, the promise of Graded Transformers lies in their ability to unify structure and learning. They allow us to encode what we already know about a domain—degree, importance, symmetry—into a format that guides what the model will learn. In doing so, they offer a path toward interpretable, mathematically principled machine learning systems that are better aligned with the structured complexity of the real world.

References

[1] T. H. Brown, *Adaptive neural networks*, Neural Networks **1** (1988), no. 1, 165–166.

[2] L. Chen, J. Wang, and H. Zhang, *Transformer-based models for gene structure prediction in genomic sequences*, Nature Biotechnology **42** (2024), 1234–1245. Available at https://doi.org/10.1038/s41587-024-02134-5.

[3] E. Clark, T. Nguyen, and L. Smith, *Graph-based transformers for dependency parsing in multilingual corpora*, Computational Linguistics **50** (2024), 123–145. Available at https://doi.org/10.1162/coli_a_00512.

[4] A. Clingher, A. Malmendier, and T. Shaska, *Isogenies, kummer surfaces, and theta functions*, Nato science for peace and security series d: Information and communication security, 2025. Available at https://www.risat.org/pdf/2025-9.pdf.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT Press, 2016.

[6] K. Johnson, M. Lee, and S. Patel, *Attention-based pde solvers for turbulent flow simulations*, Physical Review Fluids **10** (2025), 034602. Available at https://doi.org/10.1103/PhysRevFluids.10.034602.

[7] S. Lee, J. Park, and Y. Zhang, *Predicting phase transitions in condensed matter systems using transformer models*, Physical Review B **111** (2025), 045101. Available at https://doi.org/10.1103/PhysRevB.111.045101.

[8] S. Li, Y. Zhang, and R. Patel, *Dialogue-focused transformers for intent detection in conversational systems*, Neural Computing and Applications **32** (2025), 89–102. Available at https://doi.org/10.1007/s00521-024-12345-6.

[9] X. Li, Y. Zhang, and M. Chen, *Deep learning with transformers for variant effect prediction in human genomics*, Genome Research **35** (2025), 456–468. Available at https://doi.org/10.1101/gr.279123.124.

[10] ———, *Unified benchmarks for cross-domain sequence modeling*, Nature Machine Intelligence **7** (2025), 89–102. Available at https://doi.org/10.1038/s42256-024-00890-1.

[11] J. Mello, S. Salami, E. Shaska, and T. Shaska, *Rational points and zeta functions of humbert surfaces with square determinant over f_q*, Nato science for peace and security series d:

Information and communication security, 2025. Available at https://www.risat.org/pdf/2025-7.pdf.
[12] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*, 3rd ed., Pearson, 2010.
[13] B. E. A. Saleh and M. C. Teich, *Fundamentals of photonics*, 2nd ed., Wiley, 2007.
[14] E. Shaska and T. Shaska, *Machine learning for moduli space of genus two curves and an application to isogeny based cryptography*, Journal of Algebraic Combinatorics **61** (2025), 23. Available at https://www.risat.org/pdf/2024-03.pdf.
[15] T. Shaska, *Artificial neural networks on graded vector spaces*, American Mathematical Society, 2025. Available at https://www.risat.org/pdf/2024-02.pdf.
[16] ———, *Computational aspects of weighted projective varieties* (2025). Preprint, Available at https://www.risat.org/pdf/2025-16.pdf.
[17] ———, *Computing weierstrass form of superelliptic curves* (2025). Preprint, Available at https://www.risat.org/pdf/2025-8.pdf.
[18] ———, *Diagonalizable weighted hypersurfaces* (2025). Preprint, Available at https://www.risat.org/pdf/2025-14.pdf.
[19] ———, *Finsler metric clustering in weighted projective spaces* (2025). Preprint, Available at https://www.risat.org/pdf/2025-13.pdf.
[20] ———, *Finsler metric clustering in weighted projective spaces.*, arxiv (2025).
[21] ———, *Graded Neural Networks* (2025), available at 2502.17751.
[22] ———, *Graded neural networks* (2025). Preprint, Available at https://www.risat.org/pdf/2025-5.pdf.
[23] ———, *Graded transformers: Pioneering sequence modeling with graded vector spaces* (2025). Preprint, Available at https://www.risat.org/pdf/2025-11.pdf.
[24] ———, *Gröbner bases for weighted homogenous systems* (2025). Preprint, Available at https://www.risat.org/pdf/2025-12.pdf.
[25] ———, *A mathematical framework to data fabrics* (2025). Preprint, Available at https://www.risat.org/pdf/2025-15.pdf.
[26] ———, *Optimization of vector functions using the max norm* (2025). Preprint, Available at https://www.risat.org/pdf/2025-4.pdf.
[27] T. Shaska and J. Mello, *Counting of rational points on weighted projective spaces* (2025). Preprint, Available at https://www.risat.org/pdf/2025-10.pdf.
[28] T. Shaska, J. Mello, and S. Salami, *Rational points of weighted hypersurfaces over finite fields* (2025). Preprint, Available at https://www.risat.org/pdf/2025-3.pdf.
[29] T. Shaska and E. Shaska, *Galois groups of polynomials and neurosymbolic networks* (2025). Preprint, Available at https://www.risat.org/pdf/2025-1.pdf.
[30] ———, *Neuro-symbolic learning for galois groups: A machine learning approach to polynomial solvability* (2025). Preprint, Available at https://www.risat.org/pdf/2025-2.pdf.
[31] ———, *Weighted heights and git heights*, European Journal of Mathematics (2025). Available at https://www.risat.org/pdf/2025-6.pdf.
[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need*, Advances in Neural Information Processing Systems **30** (2017). Available at https://arxiv.org/abs/1706.03762.
[33] H. Wang, J. Li, and M. Chen, *Large language models for semantic role labeling in cross-lingual settings*, Transactions of the Association for Computational Linguistics **13** (2025), 234–256. Available at https://doi.org/10.1162/tacl_a_00634.
[34] H. Wang, Z. Liu, and S. Patel, *Protein structure prediction using transformer architectures*, Bioinformatics **41** (2025), 789–802. Available at https://doi.org/10.1093/bioinformatics/btab123.
[35] L. Wang, H. Chen, and J. Kim, *Transformer-based analysis of cosmological time-series for gravitational wave detection*, Astrophysical Journal **968** (2025), 123. Available at https://doi.org/10.3847/1538-4357/ad1234.
[36] C. Yun, S. Bhojanapalli, A. S. Rawat, S. J. Reddi, and S. Kumar, *Are transformers universal approximators of sequence-to-sequence functions?*, 2019. Available at https://arxiv.org/abs/1912.10077.
[37] Q. Zhang, S. Chen, and D. Kim, *Transformer-based classification of metagenomic sequences for microbial community analysis*, Nucleic Acids Research **53** (2025), e45. Available at https://doi.org/10.1093/nar/gkab456.

[38] X. Zhang, S. Chen, and D. Kim, *Contextual transformers for question answering on large-scale datasets*, Artificial Intelligence **345** (2025), 103876. Available at https://doi.org/10.1016/j.artint.2024.103876.

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

Email address: shaska@oakland.edu