GRADED TRANSFORMERS: A SYMBOLIC-GEOMETRIC APPROACH TO STRUCTURED LEARNING

T. SHASKA

To my parents, who never bowed, and never stopped teaching.

Abstract. We introduce the Graded Transformer framework, embedding algebraic inductive biases via grading transformations on vector spaces. Extending Graded Neural Networks (GNNs), we propose the Linearly Graded Transformer (LGT) and Exponentially Graded Transformer (EGT), which apply parameterized scaling—via fixed or learnable grading tuples, with exponential factors for EGT—to encode hierarchies in attention and representation layers. We establish rigorous guarantees, including universal approximation, reduced sample complexity via VC dimension bounds, Lipschitz continuity, and robustness to perturbations. A graded loss ensures stable optimization, with learnable grades enabling adaptive feature scaling. The framework excels in hierarchical and neuro-symbolic tasks across algebraic geometry, physics, NLP, biological sequence analysis, and emerging domains like graphs and finance, offering efficiency and interpretability over data-driven models. By integrating geometric and algebraic principles, Graded Transformers provide a principled approach to structured deep learning, with potential to reshape modeling of complex systems.

1. Introduction

Our prior work on AI over graded vector spaces in [1] and Graded Neural Networks (GNNs) in [2] established a framework for embedding algebraic structure into neural architectures, using fixed numerical grades to prioritize hierarchical features, enhancing efficiency and interpretability in domains like algebraic geometry and photonic signal processing. This paper introduces the *Graded Transformer* framework, comprising the Linearly Graded Transformer (LGT) and Exponentially Graded Transformer (EGT), which build on [1, 2] by integrating fixed-grade principles with the dynamic, context-aware learning of transformers. The Graded Transformer employs grading transformations $\mathrm{L}_{\mathbf{q}}$ (LGT) or $\mathrm{L}_{\mathbf{q},\lambda}$ (EGT) to prioritize critical features, with LGT using linear weights and EGT using exponential scaling. An extension to learnable grades enhances adaptability for complex, hierarchical data across scientific and linguistic domains.

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. The transformer architecture has revolutionized this field through self-attention, achieving state-of-the-art performance in tasks like machine translation, physical simulations, and genomic analysis. However, transformers struggle with hierarchical data structures, such as polynomial degrees in algebraic geometry, multi-scale phenomena in physics, syntactic heads in NLP, or regulatory regions in biology, requiring extensive training data to uncover domain-specific patterns; see [3]. This leads to high sample complexity, increased computational costs, and limited interpretability when hierarchical relationships are known a priori. The Graded Transformer addresses these challenges by embedding algebraic biases into attention and representation layers, prioritizing critical features without relying solely on data-driven attention.

The growing prevalence of hierarchical data, from algebraic systems to large language models (LLMs), underscores the need for models that leverage structural priors to enhance efficiency and interpretability. Unlike structured attention mechanisms or graph neural networks, which often sacrifice flexibility or require complex preprocessing (see [4] for details), the Graded Transformer, rooted in the GNN framework of [2], offers a compelling alternative. By encoding domain knowledge through grading tuples, it pursues three primary objectives:

i) *Feature Prioritization*: Highlighting significant features to reduce dependence on large datasets, enabling efficient learning in structured domains.

2020 *Mathematics Subject Classification.* 68T07, 14Q20, 68T05.

Key words and phrases. Graded Neural Networks, Transformers, Symbolic Machine Learning, Hierarchical Representation Learning, Graded Vector Spaces, Graded Algebras.

ii) *Computational Efficiency*: Leveraging structural priors to lower sample complexity, enabling faster convergence for hierarchical tasks.

iii) *Interpretability*: Encoding domain knowledge transparently via grading tuples, making the model's behavior predictable and explainable.

This paper is organized as follows: In Sec. 2 we develop the algebraic foundation of the framework by formalizing graded vector spaces, where basis vectors are assigned numerical grades that reflect structural importance. It introduces scalar actions, graded linear maps, and their invariance properties, laying the groundwork for encoding inductive biases into neural networks. The notion of grading is generalized to allow fractional, integer, or exponential scales, and used to define norms and activations compatible with the algebraic structure of the space.

Sec. 3 defines additive and multiplicative graded neurons and introduces graded activation functions and loss functions that respect the underlying grading geometry. It formalizes graded layers and networks, highlighting how scaling via grades allows prioritization of critical features. The section also introduces graded versions of ReLU and cross-entropy loss, showing how standard architectures can be adapted to operate on graded spaces with stable and interpretable behavior.

Sec. 4 presents the classical transformer architecture as a baseline, detailing its components — embedding, attention, encoder, decoder, and autoregressive generation — with an emphasis on permutation equivariance and computational complexity. It analyzes the transformer's limitations in modeling hierarchical data, referencing its high sample complexity and uniform attention behavior. These deficiencies motivate the graded extensions introduced in later sections, particularly in domains where structure is known in advance.

In Sec. 5 the Linearly Graded Transformer is defined by applying a fixed diagonal scaling transformation to the input representation before applying the standard transformer. The section proves universal approximation, derives VC dimension bounds, and shows robustness to noise. Graded attention mechanisms are introduced, and it is shown that linear grading preserves algebraic invariance, enhances attention rank, and improves training efficiency. The architecture preserves the computational complexity of standard transformers while improving structure-awareness and interpretability.

Sec. 6 extends the framework to exponentially graded transformations, enabling stronger amplification of high-grade features. It formalizes the exponential grading map and proves an analogous universal approximation theorem and robustness results. The section highlights use cases where exponential growth captures deep hierarchies, such as symbolic expressions or physical systems with scale separation. Attention expressivity is shown to be enhanced via exponential weighting, providing finer control over attention focus.

In Sec. 7 the training dynamics of graded transformers are addressed, focusing on stability, gradient flow, and optimization strategies for both fixed and learnable grades. Graded loss functions are revisited in the context of back propagation, and new training algorithms are introduced to accommodate dynamically updated grading tuples. The section also discusses regularization schemes, gradient normalization, and convergence behavior in the graded setting, providing a principled route toward training stable and efficient models.

The final Sec. 8 demonstrates the versatility of graded transformers through applications in diverse domains. In algebraic geometry, they model polynomial structures and zeta functions; in physics, they capture multi-scale and turbulent phenomena; in NLP, they highlight syntactic heads and semantic roles; in biology, they prioritize functional subsequences in proteins and DNA. Emerging domains like graph learning and symbolic reasoning are also discussed, underscoring the framework's capacity to bridge data-driven learning with mathematical structure.

The Graded Transformer stands poised to revolutionize artificial intelligence and scientific discovery by blending algebraic structure with data-driven flexibility, building on our foundational work in [1,2,5]. For LLMs, it offers a transformative approach to prioritize critical tokens or syntactic structures, reducing computational demands while enhancing performance in tasks like dialogue understanding, question answering, automated reasoning, and multi-modal learning. In scientific domains, its algebraic grounding, rooted in fixed and learnable grades, enables data-efficient solutions for challenges like computing zeta functions in algebraic geometry, modeling turbulent flows in physics, or predicting protein structures in biology, as detailed in Sec. 8. This fusion of mathematical rigor and adaptability invites readers to explore a new paradigm of structured learning, promising interpretable, efficient, and mathematically principled models for the complex realities of the world.

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 [1] which will be our main reference for this part. We use "grades" to denote the indices of grading (e.g., q_i), distinguishing them from "weights" used for neural network coefficients.

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 embedding numerical grades into attention mechanisms and loss functions, prioritizing features like high-degree terms in algebraic geometry or critical tokens in natural language processing.

Definition 2.1. *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 $V_i = \mathbb{F}e_i$ are the one-dimensional subspaces spanned by e_i, assigned a grade $q_i \in \mathbb{R}$.

In this paper, we often assume $q_i \in \mathbb{Q}_{>0}$ for positive scaling in machine learning tasks (see [1]). The grades form the **grading tuple** $\mathbf{q} = (q_0, \ldots, q_{d-1})$. 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.

We will denote by $\mathcal{V}_{\mathbf{q}}(\mathbb{F})$ a graded vector space over $\mathbb{F}$ with grading tuple $\mathbf{q}$. There is a large amount of literature on graded modules and graded vector spaces. The interested reader can follow the references from [1]. Below we present a bare minimum of terminology that is needed for this paper.

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. An element $v \in V$ is called **homogeneous of degree** i if it belongs to V_i (i.e., it has no components in other subspaces V_j for $j \neq i$). 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**.

Example 2.2. *For a $\mathbb{Q}$-graded space with grades $q_0 = 1, q_1 = 1/2$, a vector $[x_0, x_1]$ assigns higher priority to x_0, useful for scaling features in neural networks.*

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, i.e., $f(V_i) \subseteq W_i$ for all $i \in I$. Such maps are also called **homomorphisms** (or **morphisms**) of graded vector spaces or **homogeneous linear maps of degree 0**.

For a commutative monoid I (e.g., $\mathbb{N}$), a map f is **homogeneous of degree** $d \in I$ if

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

where $+$ is the monoid operation. If I is an additive group (e.g., $\mathbb{Z}$ or $\mathbb{Q}$), maps of degree $d \in I$ shift grades similarly, including negative or fractional degrees. For example, a map of degree d, where $d = -k$ for some $k \in I$, satisfies

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

reducing the grade by k. This is particularly relevant for $\mathbb{Q}$-graded spaces in machine learning, where negative or fractional grades enable flexible feature prioritization, as detailed in [1].

Let $\mathcal{V}^n_{\mathbf{q}}(\mathbb{F})$ and $\mathcal{V}^m_{\mathbf{q}'}(\mathbb{F})$ be graded vector spaces with $\mathbf{q} = (q_0, \ldots, q_{n-1})$ and $\mathbf{q}' = (r_0, \ldots, r_{m-1})$, respectively, where $q_i, r_j \in \mathbb{Q}_{>0}$. Let

$$\phi : \mathcal{V}^n_{\mathbf{q}}(\mathbb{F}) \to \mathcal{V}^m_{\mathbf{q}'}(\mathbb{F})$$

be an $\mathbb{F}$-linear map and $A = (a_{ij}) \in \mathrm{Mat}_{m \times n}(\mathbb{F})$ its associated matrix with respect to the standard bases. A map ϕ is **grade-preserving** if it is homogeneous of degree 0, i.e., $\phi(V_i) \subseteq W_i$ for all $i \in I$. Then we have the following:

Lemma 2.3. *ϕ 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, ϕ is grade-preserving if and only if $a_{ij} \neq 0$ implies $r_i = q_j$.

Proof. Assume ϕ is homogeneous of degree $d \in \mathbb{Q}$. For a basis vector $e_j \in V_j$ of grade q_j, $\phi(e_j)$ must lie in the subspace of grade $q_j + d$, i.e., $\phi(e_j) \in \bigoplus_{i:r_i=q_j+d} W_i$. Since $\phi(e_j) = \sum_{i=0}^{m-1} a_{ij} f_i$, where $A = (a_{ij})$ is the matrix of ϕ, non-zero components $a_{ij} \neq 0$ occur only when $f_i \in W_i$ has grade $r_i = q_j + d$. Thus, $a_{ij} \neq 0$ implies $r_i = q_j + d$.

Conversely, for a basis vector $e_j \in V_j$ of grade q_j, we have

$$\phi(e_j) = \sum_{i=0}^{m-1} a_{ij} f_i = \sum_{i:r_i=q_j+d} a_{ij} f_i,$$

since $a_{ij} = 0$ when $r_i \neq q_j + d$. Thus, $\phi(e_j)$ lies in $\bigoplus_{i:r_i=q_j+d} W_i$, which has grade $q_j + d$. For any $\mathbf{x} = \sum_{j=0}^{n-1} x_j e_j \in \mathcal{V}_{\mathbf{q}}^n(\mathbb{F})$, where $x_j e_j \in V_j$ has grade q_j, we have

$$\phi(\mathbf{x}) = \sum_{j=0}^{n-1} x_j \phi(e_j) = \sum_{j=0}^{n-1} x_j \sum_{i:r_i=q_j+d} a_{ij} f_i.$$

Each term $x_j \phi(e_j)$ is in $\bigoplus_{i:r_i=q_j+d} W_i$, so $\phi(\mathbf{x})$ is a sum of homogeneous components of grade $q_j + d$ for each q_j present in $\mathbf{x}$. Hence, ϕ is homogeneous of degree d.

The case, ϕ is grade-preserving (degree 0) if and only if $\phi(V_j) \subseteq W_j$, i.e., $d = 0$. Substituting $d = 0$ into the condition, we get $a_{ij} \neq 0 \implies r_i = q_j$, as required. □

Graded scalar multiplication $\star : \mathbb{F} \times \mathcal{V}_{\mathbf{q}}^n(\mathbb{F}) \to \mathcal{V}_{\mathbf{q}}^n(\mathbb{F})$, given by

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

for $\lambda \in \mathbb{F}$, is a graded linear map of degree 0.

Lemma 2.4 (Grading-Invariant Subspaces)**.** *Let $W \subseteq \mathcal{V}_{\mathbf{q}}^n(\mathbb{F})$ be an $\mathbb{F}$-linear subspace. The following are equivalent:*

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

Proof. Assume W is invariant under $\lambda \star \mathbf{x}$. For $\mathbf{x} = \sum_{j=0}^{n-1} x_j e_j \in W$, we have $\lambda \star \mathbf{x} = \sum_{j=0}^{n-1} \lambda^{q_j} x_j e_j \in W$. Since grades $q_j \in \mathbb{Q}_{>0}$ are distinct, choose $\lambda_1, \ldots, \lambda_n \in \mathbb{F}^\times$ such that $[\lambda_i^{q_j}]$ is invertible. Solving the system $\sum_{j=0}^{n-1} \lambda_i^{q_j} x_j e_j \in W$ yields $x_j e_j \in W$, so W is spanned by homogeneous vectors.

Conversely, if $W = \operatorname{span}\{x_j e_j \mid j \in J\}$, each $x_j e_j \in V_j$. For $\mathbf{x} = \sum_{j \in J} c_j (x_j e_j) \in W$, we have $\lambda \star \mathbf{x} = \sum_{j \in J} c_j \lambda^{q_j} (x_j e_j) \in W$, since each $\lambda^{q_j} (x_j e_j) \in V_j$. Thus, W is invariant. □

Corollary 2.5. *Each $V_j = \mathbb{F} e_j$ of grade $q_j \in \mathbb{Q}_{>0}$ is a maximal subspace of $\mathcal{V}_{\mathbf{q}}^n(\mathbb{F})$ invariant under $\lambda \star v = \lambda^{q_j} v$. Any proper subspace $W \subset V_j$ is not invariant unless $W = \{0\}$.*

Proof. For $v = x e_j \in V_j$, we have $\lambda \star v = \lambda^{q_j} x e_j \in V_j$, so V_j is invariant. Suppose $U \supset V_j$ is invariant and contains $u = x_j e_j + x_k e_k$, $k \neq j$. By Lemma 2.4, U is spanned by homogeneous vectors. Since $\lambda \star u = \lambda^{q_j} x_j e_j + \lambda^{q_k} x_k e_k \in U$ and $q_j \neq q_k$, choosing λ with $\lambda^{q_j} \neq \lambda^{q_k}$ implies $e_j, e_k \in U$, so $U = \mathcal{V}_{\mathbf{q}}^n(\mathbb{F})$. Thus, V_j is maximal. Since V_j is one-dimensional, any proper $W \subset V_j$ is $\{0\}$, which is invariant but trivial. □

3. Graded Neural Networks

Graded Neural Networks, introduced in [2] and building on [1], embed algebraic grading to prioritize features via attention mechanisms and loss weighting, forming the foundation for the Graded Transformer; see [2] for details.

Let $\mathcal{V}_{\mathbf{q}}^n(\mathbb{F})$ be as in the previous section with $n \geq 1$, and $\mathbf{q} = (q_0, \ldots, q_{n-1}) \in \mathbb{Q}^n$, which defines the **grades**, with $\operatorname{gr}(x_i) = q_i$.

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

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

where $w_i \in \mathbb{F}$ are **neural weights**, and $b \in \mathbb{F}$ 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, as in Sec. 2.

Alternatively, a **multiplicative graded neuron** can be defined as $\beta_{\mathbf{q}} : \mathcal{V}_{\mathbf{q}}^n \to \mathbb{F}$ 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. 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.

Remark 3.1. *We denote neural weights by w and grades by q_i, with exponents $w_i^{q_i}$ (or $(w_i x_i)^{q_i}$ in multiplicative neurons) reflecting grading, while w_i are parameters optimized during training. In mathematics,* graded spaces *are sometimes referred to as* weighted spaces*, which may cause confusion.*

A **graded network layer** is:

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

where

$$\mathbf{W} = [w_{j,i}^{q_i}] \in \mathbb{F}^{n \times n}, \quad \mathbf{b} = [b_0, \ldots, b_{n-1}]^t \in \mathbb{F}^n,$$

and ϕ preserves grading, with $\operatorname{gr}(y_j) = q_j$.

Layers using multiplicative neurons, $\phi(\mathbf{x}) = g(\prod_i ((\mathbf{W}\mathbf{x})_i)^{q_i} + \mathbf{b})$, are also possible but increase computational complexity; see [1]. Hybrid layers (additive/multiplicative) offer flexibility for diverse applications.

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

$$\phi_l(\mathbf{x}) = g_l(\mathbf{W}^l \mathbf{x} + \mathbf{b}^l)$$

applies a transformation defined by the matrix of neural weights $\mathbf{W}^l = [w_{j,i}^{q_i}]$, producing outputs $\hat{\mathbf{y}}$ and true values $\mathbf{y}$ in $\mathcal{V}_{\mathbf{q}}^n$ with grades $\operatorname{gr}(\hat{y}_i) = q_i$.

3.1. **ReLU Activation.** In classical neural networks, the rectified linear unit (ReLU) activation, defined as $\operatorname{ReLU}(x) = \max\{0, x\}$, applies a simple thresholding to promote sparsity and efficiency. However, for graded neural networks 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:

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

and

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

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 $\lambda \star \mathbf{x}$ we have

$$\begin{aligned} \operatorname{ReLU}_i(\lambda^{q_i} x_i) &= \max\{0, |\lambda^{q_i} x_i|^{1/q_i}\} = \max\{0, |\lambda||x_i|^{1/q_i}\} \\ &= |\lambda| \max\{0, |x_i|^{1/q_i}\}, = |\lambda| \operatorname{ReLU}_i(x_i), \end{aligned}$$

for $\lambda \in \mathbb{F}$. Thus,

$$\operatorname{ReLU}(\lambda \star \mathbf{x}) = |\lambda| \cdot \operatorname{ReLU}(\mathbf{x}),$$

unlike classical ReLU, where $\operatorname{ReLU}(\lambda \mathbf{x}) = \lambda \operatorname{ReLU}(\mathbf{x})$, for $\lambda > 0$ assumes degree 1 homogeneity.

Remark 3.2. *Unlike classical ReLU, which outputs 0 for negative inputs, the graded ReLU gives*

$$\operatorname{ReLU}_i(-x_i) = |x_i|^{1/q_i},$$

reducing sparsity and potentially affecting GNN training efficiency. An alternative, $\max\{0, |x_i|^{1/q_i} \operatorname{sgn}(x_i)\}$*, restores thresholding while maintaining graded scaling.*

3.2. **Exponential Graded Activation.** An alternative activation is the **exponential graded activation** defined as

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

and

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

This activation mitigates numerical instability for large q_i by inversely scaling inputs, ensuring smoother gradients and more gradual growth than ReLU_i for large positive inputs. For $\lambda \star \mathbf{x}$,

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

Thus,

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

reflecting the graded scaling.

3.3. **Graded Loss Functions.** In classical neural networks, loss functions like MSE

$$L = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2$$

treat coordinates equally. In contrast, graded losses for $\mathcal{V}_{\mathbf{q}}^n(\mathbb{F})$ with grades $\mathrm{gr}(x_i) = q_i$ prioritize errors via $\mathbf{q} = (q_0, \ldots, q_{n-1})$, enhancing efficiency in structured tasks like algebraic geometry or NLP. This prioritization is crucial for the Graded Transformer, where graded losses weight errors by algebraic degree to reduce sample complexity.

We define norm-based losses for regression (emphasizing error magnitude) and classification losses for categorical tasks, leveraging q_i for graded alignment. See [2] for details. These losses weight errors by algebraic degree, enhancing efficiency in structured data tasks.

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

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

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 assuming $q_i > 0$. This scales with grading: for $\lambda \star (\mathbf{y} - \hat{\mathbf{y}}) = (\lambda^{q_i}(y_i - \hat{y}_i))$,

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

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

$$\mathfrak{L}_{\mathrm{norm}}(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_{\mathbf{q}}^2 = \sum_{i=0}^{n-1} q_i |y_i - \hat{y}_i|^2, \tag{3}$$

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

The **homogeneous loss** leverages the homogeneous norm from Sec. 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:

$$\mathfrak{L}_{\mathrm{hom}}(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 = \left(\sum_{j=1}^{r} \|(\mathbf{y} - \hat{\mathbf{y}})_{d_j}\|_{d_j}^{2(r-j+1)}\right)^{\frac{1}{r}},$$

where r is the number of distinct grades, and $\|\cdot\|_{d_j}$ is the Euclidean norm on the subspace of grade d_j.

For example, for $\mathcal{V}_{(2,3)}$ with $d_1 = 2$, $d_2 = 3$, and $r = 2$

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

This emphasizes higher-graded errors over lower-graded ones.

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

$$\mathfrak{L}_{\mathrm{ce}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=0}^{n-1} q_i y_i \log(\hat{y}_i),$$

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

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

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

4. Standard Transformers

The transformer architecture has become a cornerstone of modern sequence modeling, excelling in tasks from natural language processing to time-series analysis [4, 6]. However, its limitations in capturing hierarchical data structures, such as nested polynomial degrees in algebraic geometry or multi-level protein folding in biological sequences, motivate the development of the graded transformer, which builds on our prior work in Graded Neural Networks [1, 2]. This section outlines the standard transformer's components—input embedding, encoder, decoder, and autoregressive generation—providing a mathematically rigorous foundation for understanding its strengths and shortcomings, setting the stage for the graded transformer's advancements in Sec. 5 and Sec. 6.

4.1. **Transformer Definition.** Let $\mathsf{Voc} = \{1, 2, \ldots, |\mathsf{Voc}|\}$ be a finite vocabulary of tokens. For $d \in \mathbb{N}$, let $\mathbb{R}^d$ be the embedding space for token representations. Define Voc^n as the set of sequences of length n over Voc. A sequence $(t_1, \ldots, t_n) \in \mathsf{Voc}^n$ is represented as a matrix in $\mathrm{Mat}_{n,d}(\mathbb{R})$ after embedding and positional encoding, as defined in subsequent subsections.

A **transformer** is a function

$$\mathcal{T}_\theta : \mathcal{S}_{\text{in}} \to \mathcal{S}_{\text{out}}, \tag{4}$$

where $\mathcal{S}_{\text{in}} = \bigcup_{n=1}^{n_{\max}} \mathsf{Voc}^n$ and $\mathcal{S}_{\text{out}} = \bigcup_{m=1}^{m_{\max}} \mathsf{Voc}^m$, with $n_{\max}$ and $m_{\max}$ being implementation-specific maximum lengths. It is parameterized by a collection of **learnable parameters** θ, mapping an input sequence $t = (t_1, \ldots, t_n) \in \mathsf{Voc}^n$ to an output sequence $s = (s_1, \ldots, s_m) \in \mathsf{Voc}^m$, 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.

4.2. **Input Embedding and Positional Encoding.** To process discrete token sequences in Voc^n through the transformer's neural architecture, input embeddings map tokens to continuous vectors, while positional encodings preserve sequence order, addressing the permutation invariance of attention mechanisms.

Define the embedding matrix $\mathbf{W}_e \in \mathrm{Mat}_{|\mathsf{Voc}|,d}(\mathbb{R})$. For a token $t_i \in \mathsf{Voc}$, its embedding is $\mathbf{x}_i = \mathbf{W}_e \cdot \mathbf{e}_{t_i} \in \mathbb{R}^d$, where $\mathbf{e}_{t_i} \in \mathbb{R}^{|\mathsf{Voc}|}$ is the one-hot vector with a 1 at index t_i and zeros elsewhere.

To incorporate positional information, define the **positional encoding** function $\mathrm{Pe} : \{1, \ldots, n_{\max}\} \to \mathbb{R}^d$, given by:

$$\mathrm{Pe}(i)_k = \begin{cases} \sin\left(\frac{i}{10000^{k/d}}\right) & \text{if } k \text{ is even}, \\ \cos\left(\frac{i}{10000^{(k-1)/d}}\right) & \text{if } k \text{ is odd}, \end{cases} \tag{5}$$

for $k = 0, \ldots, d-1$. For an input sequence $t = (t_1, \ldots, t_n) \in \mathsf{Voc}^n$, the embedded input matrix is:

$$X = [\mathbf{x}_1 + \mathrm{Pe}(1), \ldots, \mathbf{x}_n + \mathrm{Pe}(n)]^T \in \mathrm{Mat}_{n,d}(\mathbb{R}), \tag{6}$$

where the rows correspond to $\mathbf{x}_i + \mathrm{Pe}(i)$. These fixed encodings, while effective for sequential data, lack structural priors for hierarchical data, a limitation addressed by the graded transformer in Sec. 5.

4.3. **Encoder.** The encoder transforms the embedded input matrix X into contextualized representations, capturing global sequence dependencies through stacked layers of attention and transformation. The encoder is a function:

$$\mathrm{En}_\theta : \mathrm{Mat}_{n,d}(\mathbb{R}) \to \mathrm{Mat}_{n,d}(\mathbb{R}), \tag{7}$$

transforming X into a contextualized representation $Z = \mathrm{En}_\theta(X)$. It consists of N **layers**, each applying multi-head self-attention, a feed-forward network, residual connections, and layer normalization. While effective for sequential data, this structure struggles with hierarchical patterns, a limitation addressed by the graded transformer in Sec. 5.

The **softmax function smax** : $\mathrm{Mat}_{m,n}(\mathbb{R}) \to \mathrm{Mat}_{m,n}(\mathbb{R})$ normalizes attention scores into probabilities:

$$\mathbf{smax}(M)_{i,j} = \frac{\exp(m_{i,j})}{\sum_{k=1}^{n} \exp(m_{i,k})},$$

for a matrix $M = [m_{i,j}] \in \mathrm{Mat}_{m,n}(\mathbb{R})$, where the image of **smax** is

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

4.3.1. *Self-Attention Mechanism.* Self-attention computes weighted representations of input tokens by allowing each token to attend to all others in the sequence, capturing dependencies dynamically regardless of distance, unlike fixed convolutional filters. In its single-head form, it projects the input into query (Q), key (K), and value (V) matrices via learned linear transformations (parameterized by weight matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$), optimized during training to emphasize task-relevant token interactions. Multi-head attention extends this by running multiple parallel attention heads, each learning different aspects of relationships (e.g., syntactic vs. semantic), and concatenating their outputs for richer contextualization. The notation V for the value matrix, standard in transformer literature, represents features aggregated from matched keys, though care is needed to avoid conflicts with uses of V for vector spaces or volumes in later algebraic contexts (e.g., Sec. 5).

First, consider single-head self-attention for layer k. Define parameter matrices $\mathbf{W}_{Q,k}, \mathbf{W}_{K,k}, \mathbf{W}_{V,k} \in \mathrm{Mat}_{d,d}(\mathbb{R})$. For $X \in \mathrm{Mat}_{n,d}(\mathbb{R})$:

$$Q = X\mathbf{W}_{Q,k},\ K = X\mathbf{W}_{K,k},\ V = X\mathbf{W}_{V,k}. \tag{8}$$

The **attention matrix** is:

$$A_k(X) = \mathbf{smax}\left(\frac{QK^T}{\sqrt{d}}\right) V \in \mathrm{Mat}_{n,d}(\mathbb{R}). \tag{9}$$

Now, extend to 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 $k = 1, \ldots, N$, head $i = 1, \ldots, h$, define parameter matrices $\mathbf{W}_{Q,k,i}$, $\mathbf{W}_{K,k,i}$, $\mathbf{W}_{V,k,i} \in \mathrm{Mat}_{d,d_k}(\mathbb{R})$. For $X \in \mathrm{Mat}_{n,d}(\mathbb{R})$:

$$Q_i = X\mathbf{W}_{Q,k,i},\ K_i = X\mathbf{W}_{K,k,i},\ V_i = X\mathbf{W}_{V,k,i}. \tag{10}$$

The **attention for head** i is:

$$A_{k,i}(X) = \mathbf{smax}\left(\frac{Q_iK_i^T}{\sqrt{d_k}}\right) V_i \in \mathrm{Mat}_{n,d_k}(\mathbb{R}). \tag{11}$$

The **multi-head attention** is:

$$\mathrm{Mh}_k(X) = \mathrm{Ct}(A_{k,1}(X), \ldots, A_{k,h}(X))\mathbf{W}_{O,k} \in \mathrm{Mat}_{n,d}(\mathbb{R}), \tag{12}$$

where Ct concatenates head outputs along the feature dimension, and $\mathbf{W}_{O,k} \in \mathrm{Mat}_{hd_k,d}(\mathbb{R})$.

Remark 4.1. *Algebraically, multi-head attention induces a representation of the symmetric group via permutation equivariance (Lem. 4.17), treating sequences as multisets. However, this ignores graded hierarchies, such as polynomial rings where degrees require differential weighting. The LGT/EGT extend this to graded modules with degree-aware linear or exponential biases, as detailed in Sec. 5.*

4.3.2. *Properties of the Attention Matrix.* A matrix $M \in \mathrm{Mat}_{n,n}(\mathbb{R})$ is **row-stochastic** if all entries are non-negative and each row sums to 1, i.e., $M_{j,\cdot}\mathbf{1} = 1$ for all rows j, where $\mathbf{1}$ is the all-ones vector. Such matrices represent probability distributions over columns.

A symmetric matrix $M \in \mathrm{Mat}_{n,n}(\mathbb{R})$ is **positive-semidefinite (PSD)** if all its eigenvalues are non-negative, or equivalently, if $\mathbf{x}^T M\mathbf{x} \geq 0$ for all vectors $\mathbf{x} \in \mathbb{R}^n$.

The **Gram matrix** of vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subset \mathbb{R}^{d_k}$ is $G_{j,l} = \langle \mathbf{v}_j, \mathbf{v}_l \rangle$, which is always PSD by construction.

The **Gram kernel sense** refers to interpreting S as a kernel matrix in kernel methods, where the dot-product kernel $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle / \sqrt{d_k}$ induces a PSD Gram matrix if the kernel is positive definite (satisfying Mercer's theorem for valid reproducing kernel Hilbert spaces).

Proposition 4.2. *The attention matrix*

$$A_{k,i} = \mathbf{smax}(Q_iK_i^T/\sqrt{d_k})$$

for head i in layer k is row-stochastic. Additionally, the pre-softmax matrix

$$S = Q_iK_i^T/\sqrt{d_k}$$

is positive-semidefinite when $Q_i = K_i$ and the projections are orthogonal, with eigenvalues bounded by $O(n)$, where n is the sequence length.

Proof. The row-stochastic property follows from the softmax function: for each row j of $S = Q_i K_i^T/\sqrt{d_k}$,

$$A_{k,i,j,l} = \frac{\exp(S_{j,l})}{\sum_{m=1}^{n} \exp(S_{j,m})} \geq 0,$$

and the row sums to 1 by construction.

For the positive-semidefinite property of S, assume $Q_i = K_i$ (standard in self-attention with shared projections), making S symmetric. Then,

$$S_{j,l} = \langle \mathbf{q}_j, \mathbf{k}_l \rangle / \sqrt{d_k} = \langle \mathbf{q}_j, \mathbf{q}_l \rangle / \sqrt{d_k},$$

which is a scaled Gram matrix of the rows $\{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$. Gram matrices are positive-semidefinite, as $S = (1/\sqrt{d_k})P^T P$, where P has rows $\mathbf{q}_j$, and thus eigenvalues are non-negative.

Under orthogonality (i.e., $\mathbf{W}_{Q,k,i}^T \mathbf{W}_{Q,k,i} = I$, preserving norms), if rows of Q_i are unit-normalized (e.g., via prior layer normalization), the diagonals $S_{j,j} = 1/\sqrt{d_k}$, so the trace

$$tr(S) = n/\sqrt{d_k} = O(n/\sqrt{d_k}).$$

Since S is positive-semidefinite, eigenvalues λ satisfy

$$0 \leq \lambda \leq tr(S) = O(n/\sqrt{d_k}),$$

but in practice with d_k fixed (e.g., 64), it's $O(n)$. For the softmax-applied $A_{k,i}$, as a row-stochastic matrix, its eigenvalues lie in the unit disk $|\lambda| \leq 1$ by the Perron-Frobenius theorem, with $\lambda = 1$ always an eigenvalue, and others bounded below 1 approximately under the assumptions.

□

Remark 4.3. *This analysis aligns with the kernel interpretation of self-attention, where S acts as a dot-product kernel matrix, enabling efficient computation but treating tokens uniformly [7]. This limits applicability to graded hierarchies, unlike the LGT/EGT's degree-aware kernels in Sec. 5. Eigenvalue bounds ensure training stability, as analyzed in transformer dynamics [8].*

4.3.3. *Feed-Forward Network.* The feed-forward network applies non-linear transformations per token to enhance the encoder's representational capacity.

Definition 4.4 (Feed-Forward Network)**.** *The feed-forward network:*

$$\mathrm{Fnn}_k : \mathrm{Mat}_{n,d}(\mathbb{R}) \to \mathrm{Mat}_{n,d}(\mathbb{R}), \tag{13}$$

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

$$\mathrm{Fnn}_k(\mathbf{z}) = \mathbf{W}_{2,k} \cdot \mathrm{ReLU}(\mathbf{W}_{1,k}\mathbf{z} + \mathbf{b}_{1,k}) + \mathbf{b}_{2,k}, \tag{14}$$

where $\mathbf{W}_{1,k} \in \mathrm{Mat}_{d,d_f}(\mathbb{R})$, $\mathbf{W}_{2,k} \in \mathrm{Mat}_{d_f,d}(\mathbb{R})$, $\mathbf{b}_{1,k} \in \mathbb{R}^{d_f}$, $\mathbf{b}_{2,k} \in \mathbb{R}^d$, $d_f \in \mathbb{N}$.

Remark 4.5. *The feed-forward network introduces nonlinearity and expands the dimensionality to $d_f > d$, allowing richer representations. In graded transformers, this can be adapted to respect grading, as detailed in Sec. 5.*

Lemma 4.6 (Feed-Forward Lipschitz Continuity)**.** *The feed-forward network Fnn_k is Lipschitz continuous with constant bounded by $\|W_{1,k}\|_2 \| W_{2,k}\|_2$, assuming ReLU activation.*

Proof. ReLU is 1-Lipschitz ($|\mathrm{ReLU}(a) - \mathrm{ReLU}(b)| \leq |a - b|$), and linear transformations have Lipschitz constants equal to their spectral norms. By the chain rule, the composition's constant is the product of the individual constants. □

4.3.4. *Layer Normalization.* Layer normalization stabilizes training by normalizing token features, enabling deeper networks and smoother optimization.

Definition 4.7 (Layer Normalization)**.** *Layer normalization:*

$$\mathrm{Ln} : \mathrm{Mat}_{n,d}(\mathbb{R}) \to \mathrm{Mat}_{n,d}(\mathbb{R}), \tag{15}$$

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

$$\mathrm{Ln}(\mathbf{z}) = \frac{\mathbf{z} - \mu}{\sqrt{\sigma^2 + \epsilon}} \cdot \boldsymbol{\gamma} + \boldsymbol{\beta}, \tag{16}$$

where $\mu = \frac{1}{d}\sum_{j=1}^{d} z_j$, $\sigma^2 = \frac{1}{d}\sum_{j=1}^{d}(z_j - \mu)^2$, $\epsilon > 0$, and $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^d$.

Lemma 4.8. *Post-normalization, rows approximately satisfy*

$$\|\operatorname{Ln}(\mathbf{z}) - \boldsymbol{\beta}\|_{\boldsymbol{\gamma}} \approx \sqrt{d}$$

in high dimensions (scaled ℓ^2-norm with $\boldsymbol{\gamma}$).

Proof. After normalization, $(\mathbf{z} - \mu)/\sqrt{\sigma^2 + \epsilon}$ has mean 0 and variance 1 across features, with ℓ^2-norm approximately $\sqrt{d}$ in high dimensions by the law of large numbers. Element-wise multiplication by $\boldsymbol{\gamma}$ scales the norm, and the shift by $\boldsymbol{\beta}$ centers it, resembling a projection onto $\mathbb{S}^{d-1}$ scaled by $\sqrt{d}$ when $\boldsymbol{\gamma} = \mathbf{1}$, $\boldsymbol{\beta} = \mathbf{0}$. □

Remark 4.9. *This resembles a projection onto $\mathbb{S}^{d-1}$ scaled by $\sqrt{d}$ when $\boldsymbol{\gamma} = \mathbf{1}$, $\boldsymbol{\beta} = \mathbf{0}$, analogous to the structure of graded vector spaces in LGT/EGT (Sec. 5), where normalization aligns with hierarchical scaling.*

Example 4.10. *For a token embedding $\mathbf{z} = [0.5, -0.2, 0.3]$ with $d = 3$, if $\mu = 0.2$, $\sigma^2 \approx 0.11$, $\epsilon = 10^{-5}$, $\boldsymbol{\gamma} = \mathbf{1}$, $\boldsymbol{\beta} = \mathbf{0}$, then $\operatorname{Ln}(\mathbf{z}) \approx [0.95, -0.38, 0.57]$, with $\|\operatorname{Ln}(\mathbf{z})\|_2 \approx \sqrt{3}$, stabilizing gradients for attention layers.*

4.3.5. *Encoder Layer Composition.* The encoder layer integrates multi-head self-attention, feed-forward transformations, residual connections, and layer normalization to produce contextualized token representations while preserving gradient flow.

The k-th encoder layer is:

$$\operatorname{EnL}_k(X) = \operatorname{Ln}(X' + \operatorname{Fnn}_k(X')), \quad X' = \operatorname{Ln}(X + \operatorname{Mh}_k(X)), \tag{17}$$

where Mh_k is multi-head self-attention (Sec. 4.3), Fnn_k is the feed-forward network, and Ln is layer normalization. The encoder, with N layers, is the composition:

$$\operatorname{En}_\theta(X) = \operatorname{EnL}_N \circ \operatorname{EnL}_{N-1} \circ \cdots \circ \operatorname{EnL}_1(X). \tag{18}$$

Remark 4.11. *In graded transformers, this structure is adapted by applying grading transformations to prioritize hierarchical features, enhancing efficiency for structured data (cf. Sec. 5).*

4.4. **Limitations for Hierarchical Data.** Transformers require extensive data to learn hierarchies (e.g., syntactic trees or polynomial gradings), as attention lacks priors—provably $O(e^L)$ sample complexity for depth L via VC bounds [3].

Recent mathematical analyses model transformers as particle systems evolving on spheres, where token representations (particles) cluster over time to capture dependencies. However, this clustering is slow in high-dimensional or hierarchical settings, as quantified in the following result from [8].

Theorem 4.12. *In standard transformers modeled as particle systems on spheres, tokens cluster at an exponential rate $O(e^{-\lambda t})$ with $\lambda = O(1/(ne^{2\beta}))$, implying slow convergence (near $O(1/t)$) for large β in high-dimensional hierarchical settings.*

Here, t represents training time (or iterations), n is sequence length, and β is an inverse temperature parameter controlling clustering strength. For hierarchical data with large effective β (e.g., deep syntactic trees or multi-scale physics simulations), convergence slows dramatically, as the rate λ diminishes exponentially. This underscores transformers' inefficiency without structural priors.

4.5. **Decoder.** The decoder generates output sequences by integrating encoder outputs with causal self-attention and cross-attention. It is a function:

$$\operatorname{Dc}_\theta : \operatorname{Mat}_{t,d}(\mathbb{R}) \times \operatorname{Mat}_{n,d}(\mathbb{R}) \to \operatorname{Mat}_{t,d}(\mathbb{R}), \tag{19}$$

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

Definition 4.13 (Masked Multi-Head Self-Attention)**.** *For $Y \in \operatorname{Mat}_{t,d}(\mathbb{R})$, let*

$$Q_i = Y\mathbf{W}_{Q,l,i},\ K_i = Y\mathbf{W}_{K,l,i},\ V_i = Y\mathbf{W}_{V,l,i} \in \operatorname{Mat}_{t,d_k}(\mathbb{R}), \tag{20}$$

*with $\mathbf{W}_{Q,l,i}, \mathbf{W}_{K,l,i}$, and $\mathbf{W}_{V,l,i} \in \operatorname{Mat}_{d,d_k}(\mathbb{R})$. The **masked attention** is defined as*

$$A_{l,i}(Y) = \mathbf{smax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}} \cdot M\right) V_i, \tag{21}$$

*where $M \in \operatorname{Mat}_{t,t}(\mathbb{R})$, $M_{i,j} = 1$ if $j \le i$, and $M_{i,j} = -\infty$ if $j > i$ (implemented via masking before softmax). Then the **masked multi-head self-attention** is defined as*

$$\operatorname{Mmh}_l(Y) = \operatorname{Ct}(A_{l,1}(Y), \ldots, A_{l,h}(Y))\mathbf{W}_{O,l} \in \operatorname{Mat}_{t,d}(\mathbb{R}). \tag{22}$$

Next we define the **cross-attention** $A_{l,i}(Y,Z)$ as follows, using projections to compute the output:

Definition 4.14 (Cross-Attention)**.** *For $Y \in \mathrm{Mat}_{t,d}(\mathbb{R})$, $Z \in \mathrm{Mat}_{n,d}(\mathbb{R})$, the cross-attention is defined using the following projections to compute the output:*

$$\begin{aligned} Q_i &= Y\mathbf{W}_{Q,l,i} \in \mathrm{Mat}_{t,d_k}(\mathbb{R}), \\ K_i &= Z\mathbf{W}_{K,l,i}, \\ V_i &= Z\mathbf{W}_{V,l,i} \in \mathrm{Mat}_{n,d_k}(\mathbb{R}), \\ A_{l,i}(Y,Z) &= \mathbf{smax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \in \mathrm{Mat}_{t,d_k}(\mathbb{R}), \\ \mathrm{Ca}_l(Y,Z) &= \mathrm{Ct}(A_{l,1}(Y,Z),\ldots,A_{l,h}(Y,Z))\mathbf{W}_{O,l}. \end{aligned} \tag{23}$$

The ***l*-th decoder** layer is defined as

$$\mathrm{Dcl}_l(Y,Z) = \mathrm{Ln}(Y_l^{(2)} + \mathrm{Fnn}_l(Y_l^{(2)})), \tag{24}$$

where

$$\begin{aligned} Y_l^{(1)} &= \mathrm{Ln}(Y + \mathrm{Mmh}_l(Y)), \\ Y_l^{(2)} &= \mathrm{Ln}(Y_l^{(1)} + \mathrm{Ca}_l(Y_l^{(1)}, Z)). \end{aligned}$$

The **decoder** is defined as

$$\mathrm{Dc}_\theta(Y,Z) = \mathrm{Dcl}_N \circ \cdots \circ \mathrm{Dcl}_1(Y,Z). \tag{25}$$

Remark 4.15. *Masking in self-attention ensures causality, preventing future token leakage. Structural priors, like those in algebraic frameworks, can enhance masking by weighting tokens based on their hierarchical importance, as explored in Sec. 5.*

4.6. **Autoregressive Generation.** The transformer's ability to generate coherent sequences token by token is a cornerstone of its success in applications like language modeling and sequence translation [4]. Autoregressive generation leverages the encoder's contextualized representations to produce output sequences iteratively, a process that excels in capturing sequential dependencies but falters when faced with hierarchical data structures, requiring $O(e^L)$ samples for depth L hierarchies. This subsection formalizes the standard transformer's generation mechanism, providing a clear baseline for the trailblazing advancements introduced by the graded transformer in Sec. 6.

For an input sequence $t \in \mathsf{Voc}^n$, compute the embedded input matrix $X \in \mathrm{Mat}_{n,d}(\mathbb{R})$ as in Sec. 4.2 and the encoder output $Z = \mathrm{En}_\theta(X)$. Initialize with a start token $s_0 \in \mathsf{Voc}$, setting the initial output sequence:

$$Y_{<1} = [\mathbf{y}_0 + \mathrm{Pe}(1)]^T \in \mathrm{Mat}_{1,d}(\mathbb{R}), \quad \mathbf{y}_0 = \mathbf{W}_e \cdot e_{s_0}. \tag{26}$$

Generation proceeds recursively: $Z_{\mathrm{dec},t} = \mathrm{Dc}_\theta(Y_{<t}, Z)$, $\mathbf{z}_t = (Z_{\mathrm{dec},t})_{t,:}$, $s_t = \arg\max \mathbf{smax}(\mathbf{W}_e^T \mathbf{z}_t)$, terminating at EOS or $m_{\max}$.

This process produces the output sequence $\mathcal{T}_\theta(t) = (s_1,\ldots,s_m)$, enabling flexible sequence generation. However, its reliance on unstructured attention limits its ability to prioritize hierarchical features, a challenge overcome by the graded transformer's algebraic framework, rooted in fixed grades from [1,2] with an extension to learnable grades (Sec. 7.4), as detailed in Sec. 6.

Lemma 4.16. *The transformer $\mathcal{T}_\theta : \mathcal{S}_{in} \to \mathcal{S}_{out}$ is a well-defined function.*

Proof. Being well-defined follows from determinism of matrix operations and softmax normalization. □

This framework, while pivotal for sequence modeling, lacks structural biases for hierarchical data, a limitation addressed by the graded transformer in Sec. 6, building on [1,2] to incorporate algebraic priors for enhanced efficiency and interpretability.

Lemma 4.17 (Permutation-Equivariance)**.** *The self-attention operator:*

$$A_{l,i}(X) = \mathbf{smax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i, \tag{27}$$

is permutation-equivariant. For any permutation matrix $P \in \mathrm{Mat}_n(\mathbb{R})$, if: $Q_i' = PQ_i$, $K_i' = PK_i$, $V_i' = PV_i$, then:

$$A_{l,i}(Q_i', K_i', V_i') = PA_{l,i}(Q_i, K_i, V_i). \tag{28}$$

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

$$Q_i' K_i'^T = P Q_i K_i^T P^T = P S P^T,$$

and $\mathbf{smax}(PSP^T) = P \cdot \mathbf{smax}(S) \cdot P^T$. Thus,

$$\begin{aligned} A_{l,i}(Q_i', K_i', V_i') &= \mathbf{smax}(PSP^T)(PV_i) = P\,\mathbf{smax}(S)\,V_i \\ &= P A_{l,i}(Q_i, K_i, V_i). \end{aligned} \tag{29}$$

This completes the proof. □

Proposition 4.18 (Computational Complexity)**.** *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, $O(n^2 d)$. Quadratic n^2 arises from full pairwise interactions; graded attention reduces to $O(nd)$ via sparse hierarchies.*

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 $\mathbf{smax}(Q_i K_i^T/\sqrt{d_k})V_i$ 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 $O(ndd_f)$. □

The computational cost of attention, while significant, is balanced by careful scaling of attention scores to ensure numerical stability, as detailed below.

Proposition 4.19 (Scaling Factor)**.** *Let $Q, K \in \mathbb{R}^{n \times d_k}$, where each row $q_i, k_j \in \mathbb{R}^{d_k}$ consists of independent entries sampled from $\mathcal{N}(0,1)$. For the dot-product attention matrix $S \in \mathbb{R}^{n \times n}$, defined by:*

$$S_{i,j} = \frac{q_i \cdot k_j}{\sqrt{d_k}}, \tag{30}$$

each entry has variance $\mathrm{Var}(S_{i,j}) = 1$. This unit variance prevents saturation in softmax for high d.

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 $Q_{i,\ell}, K_{j,\ell} \sim \mathcal{N}(0,1)$, assuming random initialization with independent entries. Then:

$$q_i \cdot k_j = \sum_{\ell=1}^{d_k} Q_{i,\ell} K_{j,\ell}. \tag{31}$$

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:

$$\mathrm{Var}(q_i \cdot k_j) = \sum_{\ell=1}^{d_k} \mathrm{Var}(Q_{i,\ell} K_{j,\ell}) = d_k. \tag{32}$$

After scaling we have $\mathrm{Var}\left(\frac{q_i \cdot k_j}{\sqrt{d_k}}\right) = \frac{d_k}{d_k} = 1$. □

5. Linearly Graded Transformers

The Linearly Graded Transformer (LGT) realizes the core motivation of graded vector spaces [1, 2] by treating vectors scaled by weights as equivalent to their normalized form $[x_0, \ldots, x_{d-1}]$, focusing on relative hierarchies without exponential sensitivity. The LGT employs direct linear weights, preserving the algebraic purity of graded spaces for stable and interpretable modeling in domains like algebraic geometry and natural language processing (NLP). This section defines the LGT, proves its mathematical properties, details its architecture, and outlines training strategies, laying the foundation for applications in Sec. 7 and Sec. 8. LGT's linear scaling ensures purity in graded modules, contrasting EGT's exponential for deep hierarchies. We begin by formalizing the LGT architecture, starting with notation and its definition.

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 a grading tuple $\mathbf{q} = (q_0, \ldots, q_{d-1})$, define weights $q_i = f(q_i)$, where $f : \mathbb{R} \to \mathbb{R}^+$ is a positive function (e.g., $f(q_i) = |q_i| + 1$). The **grading matrix** is

$$\mathrm{M}_{\mathbf{q}} = \mathrm{diag}(q_0, \ldots, q_{d-1}), \tag{33}$$

with linear map $\mathrm{L}_{\mathbf{q}} : \mathcal{V} \to \mathcal{V}$ given by $\mathrm{L}_{\mathbf{q}}(\mathbf{x}) = [q_0 x_0, \ldots, q_{d-1} x_{d-1}]^t$. Define the map

$$\begin{aligned} &\phi_{\mathbf{q}} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n, \\ &X = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \to (\mathrm{M}_{\mathbf{q}}\,\mathbf{x}_1, \ldots, \mathrm{M}_{\mathbf{q}}\,\mathbf{x}_n) \end{aligned} \tag{34}$$

Let $\mathcal{T} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ be a given a standard transformer as in Eq. (4). Now we are ready for the following definition.

Definition 5.1. *The map*

$$\begin{aligned}\Psi_{\mathbf{q}} : (\mathbb{R}^d)^n &\to (\mathbb{R}^d)^n,\\ X &\to \mathcal{T}(\phi_{\mathbf{q}}(X)) = \mathcal{T}(\mathrm{M}_{\mathbf{q}}\,\mathbf{x}_1, \ldots, \mathrm{M}_{\mathbf{q}}\,\mathbf{x}_n)\end{aligned} \tag{35}$$

is called a ***Linearly Graded Transformer*** *(LGT).*

Lemma 5.2. $\Psi_{\mathbf{q}} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ *is a well-defined map.*

Proof. The map $\phi_{\mathbf{q}}$ is well-defined, as $\mathrm{M}_{\mathbf{q}} \in \mathrm{Mat}_d(\mathbb{R})$ is a linear transformation (diagonal with positive entries q_i), and applying $\mathrm{M}_{\mathbf{q}}\,\mathbf{x}_i \in \mathbb{R}^d$ to each $\mathbf{x}_i$ yields a sequence $(\mathrm{M}_{\mathbf{q}}\,\mathbf{x}_i)_{i=1}^n \in (\mathbb{R}^d)^n$. The standard transformer $\mathcal{T}$ is well-defined (Lem. 4.16). Thus, the composition

$$\Psi_{\mathbf{q}} = \mathcal{T} \circ \phi_{\mathbf{q}}$$

maps $X \in (\mathbb{R}^d)^n$ to a unique $Y \in (\mathbb{R}^d)^n$, ensuring that $\Psi_{\mathbf{q}}$ is well defined. □

This composition embeds grading into the transformer, enabling hierarchical prioritization while preserving computational efficiency.

To embed algebraic inductive biases rigorously, we situate the LGT within the category of graded vector spaces with weighted scalar action, extending the framework from Sec. 2. This categorical perspective ensures that the LGT preserves structural properties like equivariance and invariance, crucial for hierarchical data in domains such as algebraic geometry and NLP.

Let $\mathcal{C}$ be the category of graded vector spaces with weighted scalar action, where objects are graded vector spaces $\mathcal{V}^n_{\mathbf{q}}(\mathbb{R})$ over $\mathbb{R}$ with decomposition $\mathcal{V} = \bigoplus_{j=0}^{d-1} V_j$ into weight spaces. The weighted scalar action is $\lambda \star v = \lambda^{q_j} v$ for $v \in V_j$ and $\lambda \in \mathbb{R}^\times$, with grading tuple $\mathbf{q} = (q_0, \ldots, q_{d-1}) \in \mathbb{Q}^d_{>0}$. Morphisms $f : \mathcal{V}^n_{\mathbf{q}}(\mathbb{R}) \to \mathcal{V}^m_{\mathbf{q}'}(\mathbb{R})$ are linear maps that are equivariant ($f(\lambda \star v) = \lambda \star f(v)$) and preserve weights ($f(V_j) \subseteq V'_j$ for compatible $\mathbf{q}, \mathbf{q}'$).

For sequences $X \in (\mathbb{R}^d)^n \cong \mathcal{V}^n_{\mathbf{q}}(\mathbb{R})$, the action extends component-wise: $\lambda \star X = (\lambda \star x_1, \ldots, \lambda \star x_n)$. Morphisms f extend component-wise: $f(X) = (f(x_1), \ldots, f(x_n))$, where f is a morphism on $\mathcal{V}_{\mathbf{q}}(\mathbb{R})$. Invariant subspaces $W \subseteq \mathcal{V}^n_{\mathbf{q}}(\mathbb{R})$ are those closed under the action ($\lambda \star W \subseteq W$ for all $\lambda \in \mathbb{R}^\times$).

The grading map $\phi_{\mathbf{q}} : \mathcal{V}^n_{\mathbf{q}}(\mathbb{R}) \to \mathcal{V}^n_{\mathbf{q}}(\mathbb{R})$ is defined by $\phi_{\mathbf{q}}(X) = (Mx_1, \ldots, Mx_n)$, where $M = \mathrm{M}_{\mathbf{q}} = \mathrm{diag}(q_0, \ldots, q_{d-1})$ and $q_i = f(q_i) > 0$ for a positive function f.

Since M is diagonal and acts as a scalar multiple on each weight space V_j (coordinates in V_j have the same q_j), M commutes with the action on $\mathcal{V}_{\mathbf{q}}(\mathbb{R})$: $M(\lambda \star v) = M(\lambda^{q_j} v) = \lambda^{q_j}(Mv) = \lambda \star (Mv)$ for $v \in V_j$. Thus, $\phi_{\mathbf{q}}$ commutes with the action on $\mathcal{V}^n_{\mathbf{q}}(\mathbb{R})$.

Proposition 5.3 (Algebraic Invariance)**.** *If the standard transformer $\mathcal{T}$ is equivariant under the action of the symmetric group on sequences and commutes with the weighted scalar action, then the linearly graded transformer*

$$\Psi_{\mathbf{q}} = \mathcal{T} \circ \phi_{\mathbf{q}}$$

maps graded vector spaces to graded vector spaces, preserves equivariant linear maps (in the sense that $\Psi_{\mathbf{q}} \circ f = f \circ \Psi_{\mathbf{q}}$ if f is component-wise), and maps invariant subspaces under the scalar action to invariant subspaces.

Proof. Since $\mathcal{T}$ commutes with the action and $\phi_{\mathbf{q}}$ does as well (by the scalar-diagonal structure of M), we have:

$$\begin{aligned}\Psi_{\mathbf{q}}(\lambda \star X) &= \mathcal{T}(\phi_{\mathbf{q}}(\lambda \star X)) = \mathcal{T}(\lambda \star \phi_{\mathbf{q}}(X))\\ &= \lambda \star \mathcal{T}(\phi_{\mathbf{q}}(X)) = \lambda \star \Psi_{\mathbf{q}}(X).\end{aligned}$$

Hence, $\Psi_{\mathbf{q}}$ maps graded spaces to graded spaces, preserving the decomposition (as commutation respects weight spaces and $\mathcal{T}$ fixes dimensions).

For a component-wise morphism f, f commutes with M (as scalars on subspaces), so $\phi_{\mathbf{q}} \circ f = f \circ \phi_{\mathbf{q}}$. By $\mathcal{T}$'s S_n-equivariance and uniform operations, $\mathcal{T} \circ f = f \circ \mathcal{T}$. Thus:

$$\Psi_{\mathbf{q}} \circ f = \mathcal{T} \circ \phi_{\mathbf{q}} \circ f = \mathcal{T} \circ f \circ \phi_{\mathbf{q}} = f \circ \mathcal{T} \circ \phi_{\mathbf{q}} = f \circ \Psi_{\mathbf{q}}\,.$$

For invariant $W \subseteq \mathcal{V}^n_{\mathbf{q}}(\mathbb{R})$, $\phi_{\mathbf{q}}(W)$ is invariant (by commutation). Then $\mathcal{T}(\phi_{\mathbf{q}}(W))$ is invariant (by $\mathcal{T}$'s commutation), so $\Psi_{\mathbf{q}}(W)$ is invariant. Moreover, S_n-equivariance preserves the module structure under permutations. □

This invariance underpins LGT's stability in hierarchical tasks, as explored in subsequent properties like noise robustness (Prop. 5.12).

Example 5.4 (LGT in Syntactic Parsing)**.** *Consider an NLP task of syntactic parsing for a sentence with $n = 3$ tokens (e.g., "The cat sleeps"), where each token is embedded in $\mathbb{R}^d$ with $d = 4$. Suppose the grading tuple $\mathbf{q} = (1, 0.5, 0, 0)$, prioritizing syntactic head features, and set $f(q_i) = q_i + 1$, so $q_i = (2, 1.5, 1, 1)$. The sequence $\phi_{\mathbf{q}}(X)$ is fed to $\mathcal{T}$, which applies attention to prioritize the verb "sleeps" (highest in dimension 0, weighted by $q_0 = 2$). This focuses learning on syntactic heads, maintaining stability due to linear weights and normalization, aligning with NLP applications (Sec. 8).*

5.1. **LGT Properties.** Extending the algebraic invariance from Prop. 5.3, the LGT inherits key properties from linear grading, including stability and universality. By leveraging linear grading to prioritize features, it maintains expressivity through the underlying transformer while aligning with the equivalence principle of graded vector spaces from Sec. 2. These properties are crucial for reducing sample complexity in structured tasks (Sec. 8).

Lemma 5.5 (Properties of Linear Grading)**.** *Let $\mathbf{q}_{\max} = \max_i q_i$. The transformation $\mathrm{L}_{\mathbf{q}} : \mathbb{R}^d \to \mathbb{R}^d$, given by $\mathrm{L}_{\mathbf{q}}(\mathbf{x}) = \mathrm{M}_{\mathbf{q}}\,\mathbf{x}$, satisfies:*

i) Invertibility*: For $q_i > 0$, $\mathrm{L}_{\mathbf{q}}$ is invertible.*
ii) Scaling*: For $\mu \in \mathbb{R}$, $\mathrm{L}_{\mathbf{q}}(\mu\mathbf{x}) = \mu\,\mathrm{L}_{\mathbf{q}}(\mathbf{x})$.*
iii) Norm Bound*: For $\mathbf{x} \in \mathbb{R}^d$, assuming $q_i \geq 0$,*

$$\|\,\mathrm{L}_{\mathbf{q}}(\mathbf{x})\|_2 \leq \mathbf{q}_{\max}\|\mathbf{x}\|_2.$$

iv) Spectral Norm*: The eigenvalues of $\mathrm{M}_{\mathbf{q}}$ are q_i, with $\|\,\mathrm{M}_{\mathbf{q}}\,\|_2 = \mathbf{q}_{\max}$.*
v) Lipschitz Continuity*: The mapping $\mathbf{x} \mapsto \mathrm{M}_{\mathbf{q}}\,\mathbf{x}$ has constant $\mathbf{q}_{\max}$.*

Proof. Since $\mathrm{M}_{\mathbf{q}}$ is diagonal with entries $q_i \geq 0$ (or > 0 where noted):
i) The inverse is $\mathrm{M}_{\mathbf{q}}{}^{-1} = \mathrm{diag}(1/q_0, \ldots, 1/q_{d-1})$, as $\mathrm{M}_{\mathbf{q}}\,\mathrm{M}_{\mathbf{q}}{}^{-1} = I$.
ii) $\mathrm{M}_{\mathbf{q}}(\mu\mathbf{x}) = (q_0(\mu x_0), \ldots, q_{d-1}(\mu x_{d-1})) = \mu\,\mathrm{M}_{\mathbf{q}}\,\mathbf{x}$.
iii) $\|\,\mathrm{M}_{\mathbf{q}}\,\mathbf{x}\|_2 = \sqrt{\sum q_i^2 x_i^2} \leq \sqrt{\mathbf{q}_{\max}^2 \sum x_i^2} = \mathbf{q}_{\max}\|\mathbf{x}\|_2$, since $q_i \leq \mathbf{q}_{\max}$.
iv) Eigenvalues are q_i; spectral norm $\|\,\mathrm{M}_{\mathbf{q}}\,\|_2 = \max q_i = \mathbf{q}_{\max}$.
v) $\|\,\mathrm{M}_{\mathbf{q}}(\mathbf{x} - \mathbf{y})\|_2 \leq \mathbf{q}_{\max}\|\mathbf{x} - \mathbf{y}\|_2$ by iii. □

These bounds ensure grading amplifies features without explosion, key for hierarchical data like polynomials [9].

Example 5.6. *For $\mathbf{q} = (2, 1)$, $\mathrm{L}_{\mathbf{q}}$ scales the first feature 2x, bounding norms by 2—useful for prioritizing syntactic heads in NLP (cf. Sec. 8).*

Lemma 5.7 (Feature Concentration)**.** *For $\mathbf{x} \in \mathbb{R}^d$ with components sorted by q_i descending, the graded vector $\mathrm{L}_{\mathbf{q}}(\mathbf{x})$ suppresses low-grade features by $O(q_j/\mathbf{q}_{\max})$ for $q_j < \mathbf{q}_{\max}$, concentrating mass on high-grade dimensions.*

Proof. From Lem. 5.5 (norm bound), the relative contribution of component j is $\frac{q_j|x_j|}{\|\,\mathrm{L}_{\mathbf{q}}(\mathbf{x})\|_2} \leq \frac{q_j}{\mathbf{q}_{\max}}\frac{|x_j|}{\|\mathbf{x}\|_2} = O(q_j/\mathbf{q}_{\max})$, assuming bounded $\mathbf{x}$. □

To establish the LGT's approximation power for hierarchical functions, we first recall Sobolev spaces, which quantify smoothness essential for parameter-efficient learning in graded domains. This enables extending standard transformer universality with reduced complexity via grading; see [3].

Definition 5.8 (Sobolev Space)**.** *Let $\Omega \subset (\mathbb{R}^d)^n$ be a compact set with positive Lebesgue measure. The **Sobolev space**, denoted by $\mathbf{W}^{k,2}(\Omega)$, consists of functions $f : \Omega \to (\mathbb{R}^d)^n$ whose weak partial derivatives $\partial^\alpha f$ exist for all multi-indices α with $|\alpha| \leq k$, and satisfy:*

$$\|f\|_{\mathbf{W}^{k,2}} = \left(\sum_{|\alpha|\leq k} \int_\Omega \|\partial^\alpha f(X)\|_2^2\, dX\right)^{1/2} < \infty, \tag{36}$$

where $\|\cdot\|_2$ is the Euclidean norm on $(\mathbb{R}^d)^n$.

Beyond continuity, Sobolev regularity quantifies smoothness, enabling precise parameter bounds for approximation in graded settings.

Definition 5.9 (Universal Approximator)**.** *A function class $\mathcal{F}$ is a **universal approximator** for $C(\Omega, (\mathbb{R}^d)^n)$ if, for any $f \in C(\Omega, (\mathbb{R}^d)^n)$ and $\epsilon > 0$, there exists $\hat{f} \in \mathcal{F}$ such that:*

$$\sup_{X\in\Omega} \|\hat{f}(X) - f(X)\|_2 < \epsilon. \tag{37}$$

Theorem 5.10 (Universal Approximation)**.** *For a compact set $\Omega \subset (\mathbb{R}^d)^n$, the linear grading transformer $\Psi_{\mathbf{q}}$ is a universal approximator for $C(\Omega, (\mathbb{R}^d)^n)$.*

Moreover, for $f \in \mathbf{W}^{k,2}(\Omega)$ depending primarily on features with $q_i \approx \mathbf{q}_{\max}$, it requires $O(\epsilon^{-2/k}\, \mathrm{d}_{\text{eff}}\, /d)$ parameters to achieve error $\epsilon > 0$, where

$$\mathrm{d}_{\text{eff}} = |\{i : q_i \geq \mathbf{q}_{\max} - \delta\}|, \quad \delta > 0 \tag{38}$$

is the effective dimension after grading suppresses low-weight features (Lem. 5.5, norm bound).

Proof. The standard transformer $\mathcal{T} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ is a universal approximator for $C(\Omega, (\mathbb{R}^d)^n)$ with sufficient layers N and heads h [3]. For any $f \in C(\Omega, (\mathbb{R}^d)^n)$ and $\epsilon > 0$, there exists $\mathcal{T}$ such that $\sup_{X\in\Omega} \|\mathcal{T}(X) - f(X)\|_2 < \epsilon$.

The grading map $\phi_{\mathbf{q}}$ is linear, invertible (Lem. 5.5), and continuous on compact Ω, so $\phi_{\mathbf{q}}(\Omega)$ is compact. Define $g = f \circ \phi_{\mathbf{q}}^{-1}$; since f, $\phi_{\mathbf{q}}^{-1}$ are continuous, $g \in C(\phi_{\mathbf{q}}(\Omega), (\mathbb{R}^d)^n)$. There exists $\mathcal{T}$ such that

$$\sup_{Z\in\phi_{\mathbf{q}}(\Omega)} \|\mathcal{T}(Z) - g(Z)\|_2 < \epsilon.$$

For $Z = \phi_{\mathbf{q}}(X)$, $\Psi_{\mathbf{q}}(X) = \mathcal{T}(Z) \approx g(Z) = f(X)$, so

$$\sup_{X\in\Omega} \|\, \Psi_{\mathbf{q}}(X) - f(X)\|_2 < \epsilon,$$

proving universality.

For $f \in \mathbf{W}^{k,2}(\Omega)$, $\mathcal{T}$ requires $O(\epsilon^{-2/k}d)$ parameters. With focus on high q_i, d_{eff} as above; low weights contribute $O(q_i/\mathbf{q}_{\max})$, reducing parameters to $O(\epsilon^{-2/k}\, \mathrm{d}_{\text{eff}}\, /d)$, analogous to sparsity in [1]. □

Example 5.11 (LGT Approximation in Polynomials)**.** *For $\Omega = [0,1]^d$, $f(X)$ a degree-3 polynomial, $\mathbf{q} = (0,1,2,3)$, $\mathrm{d}_{\text{eff}} = 2$ ($\delta = 1$), LGT approximates f with fewer parameters than $\mathcal{T}$, focusing high-degree terms for factorization [9].*

We next examine LGT's robustness to input noise, leveraging the bounded norms from Lem. 5.5.

Proposition 5.12 (Noise Robustness)**.** *For noise $\Delta \in (\mathbb{R}^d)^n$, $\|\Delta\|_2 \leq \epsilon$, the LGT error satisfies:*

$$\|\, \Psi_{\mathbf{q}}(X + \Delta) - \Psi_{\mathbf{q}}(X)\|_2 \leq \Lambda\epsilon, \tag{39}$$

where $\Lambda \leq L_{\mathcal{T}} \cdot \mathbf{q}_{\max}^{Nh+3}$, and $L_{\mathcal{T}}$ is the Lipschitz constant of $\mathcal{T}$.

Proof. For $\Delta = (\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_n)$, $\|\Delta\|_2 = \sqrt{\sum \|\boldsymbol{\delta}_i\|_2^2} \leq \epsilon$, compute

$$\begin{aligned} \|\, \Psi_{\mathbf{q}}(X + \Delta) - \Psi_{\mathbf{q}}(X)\|_2 &= \|\mathcal{T}(\phi_{\mathbf{q}}(X + \Delta)) - \mathcal{T}(\phi_{\mathbf{q}}(X))\|_2 \\ &\leq L_{\mathcal{T}}\|\phi_{\mathbf{q}}(X + \Delta) - \phi_{\mathbf{q}}(X)\|_2. \end{aligned}$$

Since $\phi_{\mathbf{q}}(X) = (\mathrm{M}_{\mathbf{q}}\, \mathbf{x}_i)_{i=1}^n$, then

$$\begin{aligned} \|\phi_{\mathbf{q}}(X + \Delta) - \phi_{\mathbf{q}}(X)\|_2 &= \|(\mathrm{M}_{\mathbf{q}}(\mathbf{x}_i + \boldsymbol{\delta}_i) - \mathrm{M}_{\mathbf{q}}\, \mathbf{x}_i)_{i=1}^n\|_2 \\ &= \sqrt{\sum_{i=1}^n \|\, \mathrm{M}_{\mathbf{q}}\, \boldsymbol{\delta}_i\|_2^2}. \end{aligned}$$

By Lem. 5.5, $\|\, \mathrm{M}_{\mathbf{q}}\, \boldsymbol{\delta}_i\|_2 \leq \mathbf{q}_{\max}\|\boldsymbol{\delta}_i\|_2$, so

$$\begin{aligned} \sqrt{\sum \|\, \mathrm{M}_{\mathbf{q}}\, \boldsymbol{\delta}_i\|_2^2} &\leq \sqrt{\sum (\mathbf{q}_{\max}\|\boldsymbol{\delta}_i\|_2)^2} \\ &= \mathbf{q}_{\max}\sqrt{\sum \|\boldsymbol{\delta}_i\|_2^2} \leq \mathbf{q}_{\max}\epsilon. \end{aligned}$$

Thus, $\|\phi_{\mathbf{q}}(X+\Delta) - \phi_{\mathbf{q}}(X)\|_2 \leq \mathbf{q}_{\max}\epsilon$. The attention mechanism (Def. 5.13) applies $\mathrm{M}_{\mathbf{q}}{}^i$ per head, each with norm $\leq q_{i,\max} \leq \mathbf{q}_{\max}$. With N layers and h heads, plus input, feed-forward, and output layers, the LGT's Lipschitz constant is bounded by

$$\Lambda \leq L_{\mathcal{T}}\mathbf{q}_{\max}^{Nh+3}, \tag{40}$$

yielding $\|\, \Psi_{\mathbf{q}}(X + \Delta) - \Psi_{\mathbf{q}}(X)\|_2 \leq \Lambda\epsilon$. This completes the proof. □

5.2. **Linearly Graded Attention.**

Definition 5.13 (Linearly Graded Attention)**.** *For $X \in \mathrm{Mat}_{n,d}(\mathbb{R})$, the linearly graded attention for head i in layer l is defined as*

$$\mathcal{A}_{\mathbf{q}_i}(Q_i, K_i, V_i) = \mathbf{smax}\left(\frac{Q_i \, \mathrm{M}_{\mathbf{q}}{}^{i} \, K_i^T}{\sqrt{d_k}}\right) V_i, \tag{41}$$

where $Q_i = X\mathbf{W}_{Q,l,i}$, $K_i = X\mathbf{W}_{K,l,i}$, $V_i = X\mathbf{W}_{V,l,i} \in \mathrm{Mat}_{n,d_k}(\mathbb{R})$, $\mathbf{W}_{Q,l,i}, \mathbf{W}_{K,l,i}, \mathbf{W}_{V,l,i} \in \mathrm{Mat}_{d,d_k}(\mathbb{R})$, $d_k = d/h$, and:

$$\mathrm{M}_{\mathbf{q}}{}^{i} = \mathrm{diag}(q_{i,0}, \ldots, q_{i,d_k-1}), \quad q_{i,j} = f(q_{i,j}). \tag{42}$$

The bilinear form is:

$$\langle \mathbf{q}_i, \mathbf{k}_j \rangle_{\mathbf{q}_i} = \mathbf{q}_i^T \, \mathrm{M}_{\mathbf{q}}{}^{i} \, \mathbf{k}_j = \sum_{k=0}^{d_k-1} q_{i,k} q_{ik} k_{jk}. \tag{43}$$

We now demonstrate LGT's expressivity in approximating arbitrary non-negative attention matrices, complementing its robustness and universality.

Proposition 5.14 (Attention Expressivity)**.** *For any matrix $A_0 \in \mathrm{Mat}_{n,d_k}(\mathbb{R})$ with non-negative entries, there exist $Q, K, V \in \mathrm{Mat}_{n,d_k}(\mathbb{R})$, $\mathbf{q}_i$, such that $\mathcal{A}_{\mathbf{q}_i}(Q, K, V)$ approximates A_0 with:*

$$\|\mathcal{A}_{\mathbf{q}_i}(Q, K, V) - A_0\|_F \leq \delta. \tag{44}$$

Proof. Let $A_0 \in \mathrm{Mat}_{n,d_k}(\mathbb{R})$ have non-negative entries. Construct A_0' by replacing zeros with $\epsilon = \delta/(2\sqrt{nd_k})$. Choose Q, K such that:

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

so $\mathbf{smax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \approx I$. Set $V = A_0'$, yielding:

$$\mathcal{A}(Q, K, V) = \mathbf{smax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \approx A_0',$$

with error $\|\mathcal{A}(Q, K, V) - A_0'\|_F \leq \delta/2$. For linearly graded attention, set $\mathbf{q}_i$ with $q_{i,j} = cj$, $c > 0$, so $q_{i,j} = f(q_{i,j})$. For small c, $\mathrm{M}_{\mathbf{q}}{}^{i} \approx I$, so $\mathcal{A}_{\mathbf{q}_i}(Q, K, V) \approx \mathcal{A}(Q, K, V)$, with error $\leq \delta/2$. Since $\|A_0' - A_0\|_F \leq \sqrt{nd_k} \cdot \epsilon = \delta/2$, the total error is

$$\|\mathcal{A}_{\mathbf{q}_i}(Q, K, V) - A_0\|_F \leq \delta.$$

This completes the proof. □

The VC dimension quantifies the expressive power of a hypothesis class, bounding sample complexity for generalization in machine learning—crucial for demonstrating LGT's efficiency over standard transformers via grading-induced dimensionality reduction.

Definition 5.15 (VC Dimension)**.** *Let $\mathcal{F}$ be a class of binary-valued functions (hypotheses) from a set X to $\{0, 1\}$. A set of points $\{x_1, \ldots, x_n\} \subset X$ is **shattered** by $\mathcal{F}$ if, for every labeling $\{y_1, \ldots, y_n\} \in \{0, 1\}^n$, there exists $f \in \mathcal{F}$ such that $f(x_i) = y_i$ for all $i = 1, \ldots, n$. The **Vapnik-Chervonenkis (VC) dimension** of $\mathcal{F}$, denoted $\mathrm{vc}(\mathcal{F})$, is the largest integer n such that there exists a set of n points in X that can be shattered by $\mathcal{F}$. If no such finite n exists, the VC dimension is infinite; see* [10].

Proposition 5.16. *Let $\mathcal{F}$ be the hypothesis class of $\Psi_{\mathbf{q}} = \mathcal{T} \circ \phi_{\mathbf{q}}$. The VC dimension satisfies*

$$\mathrm{vc}(\mathcal{F}) \leq c \cdot N \cdot h \cdot \mathrm{d_{eff}} \cdot d \log(Nh \, \mathrm{d_{eff}}), \tag{45}$$

compared to $\mathrm{vc}(\mathcal{F}_{\mathcal{T}}) \leq c \cdot Nhd^2 \log(Nhd)$ for $\mathcal{T}$, reducing sample complexity by approximately $c \cdot Nhd(d - \mathrm{d_{eff}}) \log(Nhd)$.

Proof. For $\mathcal{T}$, with N layers, h heads, and dimension d, the parameter count is $O(Nhd^2)$, yielding

$$\mathrm{vc}(\mathcal{F}_{\mathcal{T}}) \leq c \cdot Nhd^2 \log(Nhd),$$

where c is a constant [10]. For $\Psi_{\mathbf{q}}$, $\phi_{\mathbf{q}}$ focuses on $\mathrm{d_{eff}}$ dimensions (Lem. 5.7), reducing effective parameters to $O(Nh \, \mathrm{d_{eff}} \, d)$. Thus:

$$\mathrm{vc}(\mathcal{F}) \leq c \cdot Nh \, \mathrm{d_{eff}} \, d \log(Nh \, \mathrm{d_{eff}}).$$

The reduction is

$$\begin{aligned} \mathrm{vc}(\mathcal{F}_{\mathcal{T}}) - \mathrm{vc}(\mathcal{F}) &\leq c \cdot Nh[d^2 \log(Nhd) - \mathrm{d_{eff}} \, d \log(Nh \, \mathrm{d_{eff}})] \\ &\approx c \cdot Nhd(d - \mathrm{d_{eff}}) \log(Nhd). \end{aligned}$$

Sample complexity is

$$m \geq O\left(\frac{\mathrm{vc}(\mathcal{F})\log(1/\epsilon)+\log(1/\delta)}{\epsilon^2}\right),$$

reduced by approximately $\mathrm{d_{eff}}/d$ compared to $\mathcal{T}$. □

Grading not only bounds norms but also modulates attention rank, enabling adaptive expressivity for hierarchical features.

Proposition 5.17 (Attention Rank)**.** *For graded $\mathcal{A}_{\mathbf{q}_i}(Q_i,K_i,V_i)$ and standard $\mathcal{A}(Q_i,K_i,V_i)$ the standard attention as in Def. 5.13, the singular values of*

$$Q_i\,\mathrm{M}_{\mathbf{q}}{}^{i}\,K_i^T$$

are scaled by at most $q_{i,\max}=\max_j q_{i,j}$ compared to those of $Q_iK_i^T$. Consequently, the effective numerical rank of $\mathcal{A}_{\mathbf{q}_i}(Q_i,K_i,V_i)$ may increase or decrease depending on the singular value distribution of $Q_iK_i^T$ and the grading tuple $\mathbf{q}_i$.

Proof. Let $Q_iK_i^T = U\Sigma V^T$ be the singular value decomposition with singular values $\sigma_j > 0$. Since $\mathrm{M}_{\mathbf{q}}{}^{i} = \mathrm{diag}(q_{i,0},\ldots,q_{i,d_k-1})$ is diagonal with $\|\,\mathrm{M}_{\mathbf{q}}{}^{i}\,\|_2 = q_{i,\max}$, we have

$$\|Q_i\,\mathrm{M}_{\mathbf{q}}{}^{i}\,K_i^T\|_2 \leq \|Q_i\|_2\cdot\|\,\mathrm{M}_{\mathbf{q}}{}^{i}\,\|_2\cdot\|K_i^T\|_2 = q_{i,\max}\|Q_iK_i^T\|_2.$$

Thus, each singular value $\tilde{\sigma}_j$ of $Q_i\,\mathrm{M}_{\mathbf{q}}{}^{i}\,K_i^T$ satisfies $\tilde{\sigma}_j \leq q_{i,\max}\sigma_j$. For a vector $\mathbf{v}$, $\mathrm{M}_{\mathbf{q}}{}^{i}\,\mathbf{v}$ scales components by $q_{i,j}$. In $Q_i\,\mathrm{M}_{\mathbf{q}}{}^{i}\,K_i^T$, dimensions with $q_{i,j}\approx q_{i,\max}$ dominate, while others are suppressed by $O(q_{i,j}/q_{i,\max})$; see Lem. 5.5 (norm bound). The effective numerical rank of $\mathcal{A}_{\mathbf{q}_i}(Q_i,K_i,V_i)$, defined as the number of singular values of $\mathbf{smax}\left(\frac{Q_i\,\mathrm{M}_{\mathbf{q}}{}^{i}\,K_i^T}{\sqrt{d_k}}\right)V_i$ above a threshold $\epsilon>0$, may decrease if low-weight dimensions reduce significant singular values, or increase if high-weight dimensions amplify them. □

This adaptive rank supports LGT's use in low-rank approximations for algebraic tasks (cf. Sec. 8). Finally, we confirm that LGT maintains the computational efficiency of standard attention, ensuring scalability for hierarchical tasks.

Proposition 5.18 (Attention Complexity)**.** *The linearly graded attention mechanism $\mathcal{A}_{\mathbf{q}_i}(Q,K,V)$ has computational complexity $O(n^2d_k)$, matching the complexity of standard attention.*

Proof. Consider the computation of $\mathbf{smax}\left(\frac{Q\,\mathrm{M}_{\mathbf{q}}{}^{i}\,K^T}{\sqrt{d_k}}\right)V$, where Q, K, and V are in $\mathrm{Mat}_{n,d_k}(\mathbb{R})$ and $\mathrm{M}_{\mathbf{q}}{}^{i} = \mathrm{diag}(q_{i,0},\ldots,q_{i,d_k-1})$.

First, compute $Q\,\mathrm{M}_{\mathbf{q}}{}^{i} \in \mathrm{Mat}_{n,d_k}(\mathbb{R})$. Since $\mathrm{M}_{\mathbf{q}}{}^{i}$ is diagonal, this requires scaling each of the nd_k elements of Q by the corresponding diagonal entry of $\mathrm{M}_{\mathbf{q}}{}^{i}$, costing $O(nd_k)$ operations.

Next, compute $(Q\,\mathrm{M}_{\mathbf{q}}{}^{i})K^T \in \mathrm{Mat}_{n,n}(\mathbb{R})$. This is the matrix multiplication of an $n\times d_k$ matrix by a $d_k\times n$ matrix (K^T), where each of the n^2 entries is the dot product of two d_k-dimensional vectors, costing $O(n^2d_k)$ operations.

Then, scale the resulting $n\times n$ matrix by $\frac{1}{\sqrt{d_k}}$, which requires multiplying each of the n^2 elements by a constant, costing $O(n^2)$ operations.

Apply the softmax function to this scaled matrix, normalizing each of the n rows (each with n elements). Computing the exponential for all n^2 elements costs $O(n^2)$, and summing/normalizing each row costs $O(n)$ per row, totaling $O(n^2)$ operations.

Finally, multiply the $n\times n$ softmax matrix by $V\in\mathrm{Mat}_{n,d_k}(\mathbb{R})$, producing an $n\times d_k$ matrix. Each of the nd_k entries is the dot product of an n-dimensional row from the softmax matrix and an n-dimensional column from V, costing $O(n)$ per entry and thus $O(n^2d_k)$ operations in total. □

The above result shows that the total cost is dominated by the $O(n^2d_k)$ terms from the matrix multiplications, matching the complexity of standard attention, as the additional $O(nd_k)$ from grading is asymptotically negligible.

5.3. **Architecture.** Having established LGT's foundational properties, we now detail its architecture, embedding linear grading across components to ensure stable feature prioritization for hierarchical data.

Recall linearly graded attention from Def. 5.13, which weights queries/keys for expressivity. We now extend to the full pipeline.

Definition 5.19 (Input Representation)**.** *For $\mathbf{x}_i\in X$, the LGT input is:*

$$\mathbf{x}_i' = \mathrm{M}_{\mathbf{q}}\,\mathbf{x}_i,\quad \mathbf{x}_i'' = \frac{\mathbf{x}_i'}{\|\mathbf{x}_i'\|_2},\quad \mathbf{h}_i = \sigma(\mathbf{W}\mathbf{x}_i''+\mathbf{b}), \tag{46}$$

where σ is ReLU, $\mathbf{W} \in \mathbb{R}^{m\times d}$, $\mathbf{b} \in \mathbb{R}^m$, ensuring stability via normalization.

Example 5.20 (Input in Photonic Signals). *For signal $\mathbf{x} = [1, 0.5, 0.1]$, $\mathbf{q} = (1, 1.1, 1.2)$, $\mathrm{M}_{\mathbf{q}}\,\mathbf{x} = [1, 0.55, 0.12]$. Normalized: $\mathbf{x}'' \approx [0.871, 0.479, 0.105]$, preserving ratios for frequency tasks [11].*

Positional encodings incorporate grading to bias toward important positions.

Definition 5.21 (Positional Encoding). *LGT uses linear grading:*

$$\mathrm{Pe}'(\mathbf{t}, i) = q_{\mathbf{t}}\,\mathrm{Pe}(\mathbf{t}, i), \quad q_{\mathbf{t}} = f(\mathbf{t}) = 1 - \alpha\mathbf{t}, \quad \alpha > 0, \tag{47}$$

where $\mathrm{Pe}(\mathbf{t}, 2i) = \sin\left(\frac{\mathbf{t}}{10000^{2i/d}}\right)$, $\mathrm{Pe}(\mathbf{t}, 2i+1) = \cos\left(\frac{\mathbf{t}}{10000^{2i/d}}\right)$. Input to attention:

$$\mathbf{z}_i = \mathbf{x}_i'' + \mathrm{Pe}'(\mathbf{t}_i, \cdot), \quad \mathbf{z}_i' = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2}, \tag{48}$$

with scores $(QK^T)_{ij} = (\mathbf{z}_i'\mathbf{W}_Q)(\mathbf{z}_j'\mathbf{W}_K)^T$, $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d\times d_k}$.

Alternative variants offer flexibility for diverse hierarchies.

Definition 5.22 (Attention Variants). *Alternative attention mechanisms include:*

(1) ***Linearly Graded Scores:***

$$\mathbf{sc}_{ij} = \sum_{k=1}^{d_k} q_{i,k} q_{ik} k_{jk}, \quad \mathrm{M}_{\mathbf{q}}{}^i = \mathrm{diag}(q_{i,1}, \ldots, q_{i,d_k}). \tag{49}$$

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

$$Q' = \mathrm{M}_{\mathbf{q}}{}^i\, Q, \quad K' = \mathrm{M}_{\mathbf{q}}{}^i\, K, \quad \mathbf{sc}_{ij} = \mathbf{q}_i^T\, \mathrm{M}_{\mathbf{q}}{}^i\, \mathbf{k}_j. \tag{50}$$

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

$$\mathrm{Head}_h = \mathbf{smax}\left(\frac{(\mathrm{M}_{\mathbf{q}}{}^h\, Q_h)(\mathrm{M}_{\mathbf{q}}{}^h\, K_h)^T}{\sqrt{d_k}}\right) V_h, \tag{51}$$

with $\mathrm{M}_{\mathbf{q}}{}^h = \mathrm{diag}(q_{h,0}, \ldots, q_{h,d_k-1})$.

(4) ***Linearly Graded Values:***

$$\begin{aligned} V' &= \mathrm{M}_{\mathbf{q}}{}^i\, V, \quad \mathbf{o}_i = \sum_{j=1}^{n} \alpha_{ij} (\mathrm{M}_{\mathbf{q}}{}^i\, \mathbf{v}_j), \\ \alpha_{ij} &= \mathbf{smax}\left(\frac{\mathbf{q}_i^T \mathbf{k}_j}{\sqrt{d_k}}\right). \end{aligned} \tag{52}$$

These variants enable adaptive rank (Prop. 5.17) for diverse hierarchies. Grading extends to feed-forward and output for end-to-end consistency.

Definition 5.23 (Feed-Forward and Output Layers). *The* ***linearly graded feed-forward network*** *is:*

$$\mathrm{Fnn}'(\mathbf{x}) = \mathrm{M}_{\mathbf{q}}\,\mathrm{Fnn}(\mathbf{x}), \quad \mathbf{h}' = \frac{\mathrm{Fnn}'(\mathbf{x})}{\|\,\mathrm{Fnn}'(\mathbf{x})\|_2}, \tag{53}$$

where $\mathrm{Fnn}(\mathbf{x}) = \mathrm{ReLU}(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$, $\mathbf{W}_1 \in \mathbb{R}^{d\times d_{ff}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{ff}\times d}$, and d_{ff} is the hidden dimension. The ***linearly graded output*** *is:*

$$\mathbf{h}' = \mathrm{M}_{\mathbf{q}}\,\mathbf{h}, \quad \mathbf{z} = \mathbf{W}_{out}\mathbf{h}' + \mathbf{b}_{out}, \tag{54}$$

followed by $\mathbf{smax}(\mathbf{z})$.

Example 5.24 (LGT in Polynomial Prediction). *For a polynomial sequence with $X \in (\mathbb{R}^4)^m$, $d = 4$ (degrees 0 to 3), set $q_i = 1 + 0.5i$. The loss*

$$\mathcal{L} = \sum q_k \ell(\hat{y}_{i,k}, y_{i,k})$$

weights higher-degree terms (e.g., $q_3 = 2.5$) linearly, focusing learning on critical monomials with stable gradients.

Overall, the architecture ensures efficiency without overhead (Prop. 5.34), positioning LGT for hierarchical applications; see Sec. 8 for further details.

With graded input defined, we verify its stability, ensuring robust prioritization under noise in hierarchical tasks.

Theorem 5.25 (Input Stability). *The mapping*

$$\mathbf{x}_i \mapsto \mathbf{x}_i'' = \frac{\mathrm{M}_{\mathbf{q}}\,\mathbf{x}_i}{\|\,\mathrm{M}_{\mathbf{q}}\,\mathbf{x}_i\|_2}$$

is Lipschitz continuous with constant at most $\mathbf{q}_{\max}$*, assuming* $q_i \geq 0$.

Proof. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, let $\mathbf{x}' = \mathrm{M}_{\mathbf{q}}\,\mathbf{x}$, $\mathbf{y}' = \mathrm{M}_{\mathbf{q}}\,\mathbf{y}$. By Lem. 5.5:

$$\|\mathbf{x}' - \mathbf{y}'\|_2 \leq \mathbf{q}_{\max}\|\mathbf{x} - \mathbf{y}\|_2. \tag{55}$$

Normalization $\mathbf{z} \mapsto \frac{\mathbf{z}}{\|\mathbf{z}\|_2}$ is 1-Lipschitz for $\mathbf{z} \neq 0$. Let $\mathbf{x}'' = \frac{\mathbf{x}'}{\|\mathbf{x}'\|_2}$, $\mathbf{y}'' = \frac{\mathbf{y}'}{\|\mathbf{y}'\|_2}$. Then:

$$\|\mathbf{x}'' - \mathbf{y}''\|_2 \leq \frac{\|\mathbf{x}' - \mathbf{y}'\|_2}{\min(\|\mathbf{x}'\|_2, \|\mathbf{y}'\|_2)} \leq \frac{\mathbf{q}_{\max}\|\mathbf{x} - \mathbf{y}\|_2}{\epsilon}, \tag{56}$$

assuming $\|\mathbf{x}'\|_2, \|\mathbf{y}'\|_2 \geq \epsilon > 0$ (via regularization or non-zero min-norm, as in transformer layer norm). For normalized inputs ($\|\mathbf{x}\|_2, \|\mathbf{y}\|_2 \approx 1$), the constant is $\mathbf{q}_{\max}$. □

This implies bounded activations post-grading.

Corollary 5.26 (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 $\|\mathbf{x}_i''\|_2 = 1$ assuming $\mathbf{x}_i' \neq 0$. □

These ensure graded inputs remain bounded and stable, preventing distortion in attention for structured data. Grading also bounds Jacobians and biases positionals, ensuring stable, order-aware processing.

Lemma 5.27 (Jacobian Bound). *The Jacobian of* $\mathbf{x}_i \mapsto \mathbf{x}_i' = \mathrm{M}_{\mathbf{q}}\,\mathbf{x}_i$ *has operator norm* $\mathbf{q}_{\max}$.

Proof. The map is linear, with Jacobian $\mathrm{M}_{\mathbf{q}}$, a diagonal matrix with entries q_i. The operator norm is the maximum singular value, $\|\,\mathrm{M}_{\mathbf{q}}\,\|_2 = \max_i q_i = \mathbf{q}_{\max}$. □

Proposition 5.28 (Positional Bias). *For* $f(\mathbf{t}) = 1 - \alpha\mathbf{t}$, $\alpha > 0$, *the linearly graded attention biases earlier positions.*

Proof. The encoding is:

$$\mathrm{Pe}'(\mathbf{t}, i) = (1 - \alpha\mathbf{t})\,\mathrm{Pe}(\mathbf{t}, i). \tag{57}$$

Since $1 - \alpha\mathbf{t}$ decreases as $\mathbf{t}$ increases (for $\alpha > 0$), earlier positions (small $\mathbf{t}$) have larger weights. In attention scores:

$$\langle \mathbf{z}_i', \mathbf{z}_j' \rangle_{\mathbf{q}_i} \approx \langle \mathbf{x}_i'' + (1 - \alpha\mathbf{t}_i)\,\mathrm{Pe}(\mathbf{t}_i), \mathbf{x}_j'' + (1 - \alpha\mathbf{t}_j)\,\mathrm{Pe}(\mathbf{t}_j) \rangle_{\mathbf{q}_i}, \tag{58}$$

earlier positions contribute larger terms, biasing attention toward them. □

Lemma 5.29 (Positional Stability). *The mapping* $\mathbf{t} \mapsto \mathbf{z}_i'$ *is Lipschitz continuous with constant bounded by* $Cq_{\mathbf{t},\max}$, *where* $q_{\mathbf{t},\max} = \max_{\mathbf{t}} |f(\mathbf{t})|$.

Proof. For $\mathbf{z}_i = \mathbf{x}_i'' + \mathrm{Pe}'(\mathbf{t}, \cdot)$, $\mathbf{z}_i' = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2}$, and positions $\mathbf{t}, \mathbf{t}'$:

$$\|\mathbf{z}_i(\mathbf{t}) - \mathbf{z}_i(\mathbf{t}')\|_2 = \|\,\mathrm{Pe}'(\mathbf{t}, \cdot) - \mathrm{Pe}'(\mathbf{t}', \cdot)\|_2. \tag{59}$$

Since $\mathrm{Pe}(\mathbf{t}, i)$ is bounded ($|\,\mathrm{Pe}(\mathbf{t}, i)| \leq 1$):

$$\|\,\mathrm{Pe}'(\mathbf{t}, i) - \mathrm{Pe}'(\mathbf{t}', i)\|_2 \leq |f(\mathbf{t}) - f(\mathbf{t}')| \cdot |\,\mathrm{Pe}(\mathbf{t}, i)|. \tag{60}$$

For $f(\mathbf{t}) = 1 - \alpha\mathbf{t}$, $|f(\mathbf{t}) - f(\mathbf{t}')| = \alpha|\mathbf{t} - \mathbf{t}'|$, so:

$$\|\,\mathrm{Pe}'(\mathbf{t}, \cdot) - \mathrm{Pe}'(\mathbf{t}', \cdot)\|_2 \leq \alpha|\mathbf{t} - \mathbf{t}'|\sqrt{d}. \tag{61}$$

Normalization is 1-Lipschitz, so the constant is bounded by $Cq_{\mathbf{t},\max}$, where C depends on α and d, and $q_{\mathbf{t},\max} = \max_{\mathbf{t}} |f(\mathbf{t})|$. □

We now analyze stability of graded inputs/positionals, ensuring robust hierarchies like early bias for syntactic roots.

Theorem 5.30 (Attention Stability). *For the Linearly Graded Queries/Keys variant, the score* $\mathbf{sc}_{ij} = \mathbf{q}_i^T\,\mathrm{M}_{\mathbf{q}}{}^i\,\mathbf{k}_j$ *is Lipschitz continuous with constant at most* $q_{i,\max}C$, *where* C *bounds* $\|\mathbf{q}_i\|_2, \|\mathbf{k}_j\|_2$.

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

$$|\mathbf{q}_i^T \,\mathrm{M}_{\mathbf{q}}{}^i\, \mathbf{k}_j - \mathbf{q}_i'^T \,\mathrm{M}_{\mathbf{q}}{}^i\, \mathbf{k}'_j| \le \|\mathbf{q}_i - \mathbf{q}'_i\|_2 \|\,\mathrm{M}_{\mathbf{q}}{}^i\, \mathbf{k}_j\|_2 + \|\mathbf{q}'_i\|_2 \|\,\mathrm{M}_{\mathbf{q}}{}^i(\mathbf{k}_j - \mathbf{k}'_j)\|_2.$$

By Lem. 5.5,

$$\|\,\mathrm{M}_{\mathbf{q}}{}^i\, \mathbf{k}_j\|_2 \le q_{i,\max}\|\mathbf{k}_j\|_2,$$
$$\|\,\mathrm{M}_{\mathbf{q}}{}^i(\mathbf{k}_j - \mathbf{k}'_j)\|_2 \le q_{i,\max}\|\mathbf{k}_j - \mathbf{k}'_j\|_2.$$

With $\|\mathbf{q}'_i\|_2, \|\mathbf{k}_j\|_2 \le C$ we have

$$|\mathbf{q}_i^T \,\mathrm{M}_{\mathbf{q}}{}^i\, \mathbf{k}_j - \mathbf{q}_i'^T \,\mathrm{M}_{\mathbf{q}}{}^i\, \mathbf{k}'_j| \le q_{i,\max} C(\|\mathbf{q}_i - \mathbf{q}'_i\|_2 + \|\mathbf{k}_j - \mathbf{k}'_j\|_2).$$

□

Proposition 5.31 (Head Diversity)**.** *Distinct grading tuples* $\mathbf{q}_h$ *in the Linearly Graded Multi-Head variant enhance representational capacity.*

Proof. Each head computes

$$\mathrm{Head}_h = \mathbf{smax}\left(\frac{(\mathrm{M}_{\mathbf{q}}{}^h\, Q_h)(\mathrm{M}_{\mathbf{q}}{}^h\, K_h)^T}{\sqrt{d_k}}\right) V_h,$$

with $\mathrm{M}_{\mathbf{q}}{}^h = \mathrm{diag}(q_{h,0}, \ldots, q_{h,d_k-1})$. Distinct $\mathbf{q}_h$ produce unique $\mathrm{M}_{\mathbf{q}}{}^h$, scaling query and key dimensions differently. The singular values of $\mathrm{M}_{\mathbf{q}}{}^h\, Q_h K_h^T (\mathrm{M}_{\mathbf{q}}{}^h)^T$ vary with $\mathbf{q}_h$, projecting each head onto a distinct subspace, enhancing the concatenated output's ability to capture diverse dependencies. □

Proposition 5.32 (Feed-Forward Stability)**.** *The linearly graded feed-forward mapping* $\mathbf{x} \mapsto \mathrm{Fnn}'(\mathbf{x}) = \mathrm{M}_{\mathbf{q}}\,\mathrm{Fnn}(\mathbf{x})$ *has Lipschitz constant at most* $\mathbf{q}_{\max} L_{\mathrm{Fnn}}$.

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

$$\|\,\mathrm{Fnn}'(\mathbf{x}) - \mathrm{Fnn}'(\mathbf{y})\|_2 = \|\,\mathrm{M}_{\mathbf{q}}(\mathrm{Fnn}(\mathbf{x}) - \mathrm{Fnn}(\mathbf{y}))\|_2 \le \mathbf{q}_{\max}\|\,\mathrm{Fnn}(\mathbf{x}) - \mathrm{Fnn}(\mathbf{y})\|_2,$$

by Lem. 5.5. Since Fnn is Lipschitz with constant L_{Fnn} (due to ReLU and linear layers)

$$\|\,\mathrm{Fnn}(\mathbf{x}) - \mathrm{Fnn}(\mathbf{y})\|_2 \le L_{\mathrm{Fnn}}\|\mathbf{x} - \mathbf{y}\|_2.$$

Thus,

$$\|\,\mathrm{Fnn}'(\mathbf{x}) - \mathrm{Fnn}'(\mathbf{y})\|_2 \le \mathbf{q}_{\max} L_{\mathrm{Fnn}}\|\mathbf{x} - \mathbf{y}\|_2.$$

Normalization is 1-Lipschitz, preserving the constant. □

Proposition 5.33 (Output Stability)**.** *The output mapping* $\mathbf{h} \mapsto \mathbf{z} = W_{out}(\mathrm{M}_{\mathbf{q}}\,\mathbf{h}) + \mathbf{b}_{out}$ *is Lipschitz with constant at most* $\mathbf{q}_{\max} L_{out}$.

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

$$\|\mathbf{z}_1 - \mathbf{z}_2\|_2 = \|W_{\mathrm{out}}\,\mathrm{M}_{\mathbf{q}}(\mathbf{h}_1 - \mathbf{h}_2)\|_2 \le \|W_{\mathrm{out}}\|_2 \|\,\mathrm{M}_{\mathbf{q}}(\mathbf{h}_1 - \mathbf{h}_2)\|_2.$$

By Lem. 5.5, $\|\,\mathrm{M}_{\mathbf{q}}(\mathbf{h}_1 - \mathbf{h}_2)\|_2 \le \mathbf{q}_{\max}\|\mathbf{h}_1 - \mathbf{h}_2\|_2$. With $L_{\mathrm{out}} = \|W_{\mathrm{out}}\|_2$ we have

$$\|\mathbf{z}_1 - \mathbf{z}_2\|_2 \le \mathbf{q}_{\max} L_{\mathrm{out}}\|\mathbf{h}_1 - \mathbf{h}_2\|_2.$$

□

Proposition 5.34 (Computational Complexity)**.** *The LGT has the same asymptotic complexity as the standard transformer,* $O(n^2 d + nd^2)$.

Proof. The standard transformer's complexity is $O(n^2 d)$ for attention and $O(nd^2)$ for feed-forward layers (Prop. 4.18). Each $\mathrm{M}_{\mathbf{q}}$ or $\mathrm{M}_{\mathbf{q}}{}^i$ is a diagonal matrix multiplication, costing $O(d)$ or $O(d_k)$ per token, or $O(nd)$ for n tokens across input, positional encoding, attention, feed-forward, and output layers. Since $nd \ll n^2 d, nd^2$, the total complexity remains $O(n^2 d + nd^2)$. □

5.4. **Training and Optimization.** The Linearly Graded Transformer introduces linear grading transformations to prioritize hierarchical features, necessitating tailored training strategies to optimize its parameters effectively. Unlike our previous work [1, 2], where grades were fixed weights of the vector space, here we treat the grading tuples $\mathbf{q} = (q_1, \ldots, q_d)$ and $\mathbf{q}_i = (q_{i,1}, \ldots, q_{i,d_k})$ as learnable parameters, optimized via gradient descent to adaptively prioritize features based on data. This enhances flexibility but introduces challenges in optimization complexity. This section formalizes the linearly graded loss function, develops regularization and optimization techniques, establishes convergence and stability guarantees, and outlines LGT-specific training considerations for structured tasks in domains like algebraic geometry and NLP [1, 2]. Linear's simplicity contrasts EGT's power for deep hierarchies.

5.4.1. *Linearly Graded Loss Function.* The linearly graded loss function embeds hierarchical priors by weighting prediction errors according to their grades, ensuring that critical features, such as high-degree polynomial terms or syntactic heads, contribute more significantly to the optimization objective.

Definition 5.35. *For an output sequence $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_m) \in (\mathbb{R}^d)^m$, with predicted outputs $\hat{y}_{i,k}$ and true outputs $y_{i,k} \in \mathbb{R}$ for the k-th dimension of token i, the **linearly graded loss** function is:*

$$\mathcal{L} = \sum_{i=1}^{m} \sum_{k=1}^{d} q_k \, \ell(\hat{y}_{i,k}, y_{i,k}), \tag{62}$$

where:

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

Example 5.36 (Linearly Graded Loss in Polynomial Degree Prediction)**.** *Consider a task in algebraic geometry where the LGT predicts the degrees of terms in a polynomial sequence for computing zeta functions of hypersurfaces [9]. Let the output be $Y \in (\mathbb{R}^d)^m$, with $d = 4$ representing degrees 0 to 3, and let $\mathbf{q} = (q_1, \ldots, q_4) = (0, 0.5, 1, 2)$, with $f(q_k) = q_k + 1$, so $q_k = (1, 1.5, 2, 3)$, prioritizing higher-degree terms. The loss is*

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{m} [&\ell(\hat{y}_{i,1}, y_{i,1}) + 1.5\ell(\hat{y}_{i,2}, y_{i,2}) \\ &+2\ell(\hat{y}_{i,3}, y_{i,3}) + 3\ell(\hat{y}_{i,4}, y_{i,4})] \, . \end{aligned} \tag{63}$$

For a sequence with $m = 2$, true outputs $y_{1,4} = 1$, $y_{2,4} = 1$, and predicted outputs $\hat{y}_{1,4} = 0.8$, $\hat{y}_{2,4} = 0.9$, using cross-entropy $\ell(\hat{y}, y) = -y \log \hat{y} - (1-y) \log(1-\hat{y})$, the loss term for degree 3 is amplified, reducing sample complexity by focusing on critical terms (Prop. 5.16).

5.4.2. *Regularization and Optimization.* The LGT's linear grading ensures numerical stability compared to exponential scaling, but learnable grades still require regularization to control their magnitudes and strategies to stabilize training. These methods address the model's multi-head grading challenges.

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

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \gamma \|\mathbf{q}\|_2^2, \tag{64}$$

with regularization weight $\gamma > 0$. Optimization uses standard gradient-based methods like Adam [12]. The gradient of a single attention score with respect to grade q_k in Def. 5.13 is:

$$\frac{\partial \text{Score}_{ij}}{\partial q_k} = f'(q_k) q_{ik} k_{jk}, \tag{65}$$

where q_{ik}, k_{jk} denote the k-th components of the query and key vectors at positions i and j, and $q_k = q_{i,k}$ is the attention grade for head i. For $f(q_k) = q_k + 1$, $f'(q_k) = 1$, simplifying optimization.

To optimize the LGT effectively, we propose:

i) *Gradient Clipping*: Cap the gradient norm to a threshold $\tau > 0$:

$$\mathbf{g}' = \min\left(1, \frac{\tau}{\|\mathbf{g}\|_2}\right) \mathbf{g}, \tag{66}$$

where $\mathbf{g}$ is the gradient of $\mathcal{L}_{\text{total}}$.

ii) *Warm-Starting*: Initialize $\mathbf{q}$ and $\mathbf{q}_i$ with domain-informed grades (e.g., $q_k = ck$, $c > 0$, for polynomial degrees).

iii) *Multi-Head Grade Coordination*: Regularize the variance of attention grades:

$$\mathcal{L}_{\text{coord}} = \sum_{i=1}^{h} \left\| \mathbf{q}_i - \frac{1}{h} \sum_{j=1}^{h} \mathbf{q}_j \right\|_2^2, \tag{67}$$

added to $\mathcal{L}_{\text{total}}$ with weight $\gamma_{\text{coord}} > 0$.

5.4.3. *Convergence and Gradient Stability.* Convergence and gradient stability are critical for ensuring the LGT's training process is robust, given the learnable grading transformations.

Theorem 5.37 (Convergence)**.** *Fix a grading vector* $\mathbf{q}$*. If the loss function* ℓ *has a Lipschitz continuous gradient with constant* $L_\ell > 0$*, then gradient descent with sufficiently small step size converges to a stationary point of the LGT's loss.*

Proof. The LGT $\Psi_{\mathbf{q_q}}$ is Lipschitz continuous by Prop. 5.12, with constant $\Lambda \leq L_{\mathcal{T}} \mathbf{q}_{\max}^{Nh+3}$, where $\mathbf{q}_{\max} = \max_{k=1,\dots,d} q_k$. The linearly graded loss $\mathcal{L}$ has gradient Lipschitz constant $\max_k q_k \cdot L_\ell$ w.r.t. outputs. Assuming bounded weights, $\Psi_{\mathbf{q}}$ has Lipschitz gradient w.r.t. parameters (e.g., via ReLU/softmax compositions [13]). By chain rule, the composite $\mathcal{L} \circ \Psi_{\mathbf{q}}$ has Lipschitz gradient with constant proportional to $\Lambda \cdot \max_k q_k$. Classical gradient descent results [14] ensure that step size $\eta < 2/(\Lambda \cdot \max_k q_k)$ yields convergence to a stationary point. □

Gradient stability ensures reliable updates during optimization, particularly for learnable grades where sensitivity to q_k could cause divergence.

Proposition 5.38 (Gradient Stability)**.** *Assume* $\|\mathbf{q}_i\|_2, \|\mathbf{k}_j\|_2 \leq C$ *for some constant* $C > 0$ *and that* ℓ *has a Lipschitz continuous gradient with constant* $L_\ell > 0$*. Then the gradient* $\frac{\partial \mathcal{L}_{\text{total}}}{\partial q_k}$ *is Lipschitz continuous in* q_k*, with constant proportional to* $\mathbf{q}_{\max}$*.*

Proof. The derivative is

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

For $f(q_k) = q_k + 1$, $f'(q_k) = 1$. The second derivative is

$$\frac{\partial^2 \mathcal{L}_{\text{total}}}{\partial q_k^2} = \sum_{i=1}^{m} \left(f'(q_k) \frac{\partial \ell}{\partial \hat{y}_{i,k}} \cdot \frac{\partial \hat{y}_{i,k}}{\partial q_k} + q_k \frac{\partial^2 \ell}{\partial \hat{y}_{i,k}^2} \left(\frac{\partial \hat{y}_{i,k}}{\partial q_k} \right)^2 + q_k \frac{\partial \ell}{\partial \hat{y}_{i,k}} \cdot \frac{\partial^2 \hat{y}_{i,k}}{\partial q_k^2} \right) + 2\gamma.$$

Since ℓ has Lipschitz gradient L_ℓ, its second derivative is bounded (e.g., $\left| \frac{\partial^2 \ell}{\partial \hat{y}_{i,k}^2} \right| \leq L_\ell$). The output $\hat{y}_{i,k}$ depends on q_k through the LGT layers, including the graded output layer, which is Lipschitz w.r.t. q_k with constant proportional to $\mathbf{q}_{\max}$ (from scaling in $\mathrm{M}_{\mathbf{q}}$). Assuming bounded higher derivatives in the transformer (e.g., via ReLU [13]), the dominant terms (involving $q_k \cdot (\partial \hat{y}/\partial q_k)^2$) are bounded by $O(m \mathbf{q}_{\max})$, yielding the claimed Lipschitz constant for the gradient. □

With gradient stability established, we now address overall training stability for learnable grades and sensitivity to the grading function f, which are crucial for practical optimization of the LGT.

Proposition 5.39 (Training Stability for Learned Grades)**.** *Assume* $\|\mathbf{q}_i\|_2, \|\mathbf{k}_j\|_2 \leq C$ *for some constant* $C > 0$*. Then, the training process with learned* $\mathbf{q}$ *is stable if the learning rate for* q_k *satisfies* $\eta_q < 1/\mathbf{q}_{\max}$*.*

Proof. From Prop. 5.38, the gradient $\frac{\partial \mathcal{L}_{\text{total}}}{\partial q_k}$ is Lipschitz continuous with constant $L_g \propto \mathbf{q}_{\max}$. For gradient descent updates $q_k \leftarrow q_k - \eta_q \frac{\partial \mathcal{L}_{\text{total}}}{\partial q_k}$, stability (i.e., non-divergence) requires $\eta_q < 1/L_g$. Thus, choosing $\eta_q < 1/\mathbf{q}_{\max}$ ensures bounded updates, preventing divergence due to sensitivity in q_k. □

Proposition 5.40 (Sensitivity to Grading Function)**.** *The LGT's training is sensitive to the choice of* f*, with the loss gradient's magnitude scaling as* $O(\mathbf{q}_{\max})$*. Bounding* $\mathbf{q}_{\max} \leq q_{\max,bound}$ *ensures stable optimization.*

Proof. The gradient $\frac{\partial \mathcal{L}_{\text{total}}}{\partial q_k}$ includes terms like $f'(q_k)\ell(\hat{y}_{i,k}, y_{i,k})$, which scale with $\mathbf{q}_{\max}$ (since $f'(q_k)$ is constant for linear f, but the overall amplification from grading affects sensitivity). From Prop. 5.12, the LGT's Lipschitz constant scales as $\mathbf{q}_{\max}^{Nh+3}$, where N is the number of layers. Large $\mathbf{q}_{\max}$ amplifies gradients, risking instability. Bounding $\mathbf{q}_{\max} \leq q_{\max,\text{bound}}$ (e.g., $q_{\max,\text{bound}} = 10$) limits the gradient magnitude, ensuring stable optimization with $\eta_q < 1/q_{\max,\text{bound}}$. □

5.4.4. *LGT-Specific Training Challenges.* Building on the convergence and stability results, we now address practical challenges in training the LGT, particularly for learnable grades, which enable adaptive hierarchies but require careful optimization to maintain stability.

Unlike our previous work [1, 2], where grades were fixed, here the grading tuples $\mathbf{q} = (q_1, \ldots, q_d)$ and $\mathbf{q}_i = (q_{i,1}, \ldots, q_{i,d_k})$ are optimized via gradient descent, enhancing flexibility at the cost of increased complexity. Algorithm 5.1 mitigates these issues through targeted strategies.

Algorithm 5.1 (Grade Initialization and Training)**.**

***Input**: Training data, initial parameters θ, grading tuples $\mathbf{q}$, $\mathbf{q}_i$, total steps T, regularization weights γ, γ_{coord}, gradient threshold τ.*

***Note**: Grades $\mathbf{q}$, $\mathbf{q}_i$ are learnable parameters.*

***Initialize**:*

Set $\mathbf{q} = (q_1, \ldots, q_d)$ with domain-informed grades (e.g., $q_k = c \cdot k$, $c > 0$).

Set $\mathbf{q}_i = (q_{i,1}, \ldots, q_{i,d_k})$ similarly for each attention head.

***For** $t = 1, \ldots, T$:*

Compute loss $\mathcal{L}_{\text{total}} + \gamma_{coord}\mathcal{L}_{\text{coord}}$, where:

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

$\mathcal{L}_{\text{coord}} = \sum_{i=1}^{h} \left\| \mathbf{q}_i - \frac{1}{h}\sum_{j=1}^{h} \mathbf{q}_j \right\|_2^2$.

Compute gradient $\mathbf{g}$ and clip to $\mathbf{g}' = \min\left(1, \frac{\tau}{\|\mathbf{g}\|_2}\right)\mathbf{g}$.

Update parameters using Adam with learning rate $\eta_q < 1/\mathbf{q}_{\max}$ for grades.

***Output**: Optimized parameters θ, $\mathbf{q}$, $\mathbf{q}_i$.*

The following components of Algorithm 5.1 address the primary training challenges:

i) *Grade Tuning*: Large q_k values amplify high-grade features (Prop. 5.12), but risk over-prioritization. The algorithm mitigates this by regularizing $\|\mathbf{q}\|_2^2$, ensuring balanced grades.

ii) *Multi-Head Grading*: Distinct grading tuples $\mathbf{q}_i$ per attention head enhance expressivity (Prop. 5.31) but may lead to conflicting priorities. The regularization term $\mathcal{L}_{\text{coord}}$ ensures coordinated grading across heads.

iii) *Grade Initialization*: Random initialization of $\mathbf{q}$ and $\mathbf{q}_i$ may lead to suboptimal feature prioritization. The algorithm uses domain-informed grades (e.g., $q_k = c \cdot k$) to leverage prior knowledge, improving convergence.

The LGT's training leverages its hierarchical structure to prioritize critical features, as shown in the polynomial degree prediction example. The use of learnable grades $\mathbf{q}$, $\mathbf{q}_i$, unlike fixed grades in [1, 2], enhances flexibility but increases optimization complexity, addressed by Algorithm 5.1 through Items i to iii. Empirical validation on tasks like equation parsing and symbolic regression, discussed in Sec. 8, is a key next step. Linear's simplicity contrasts EGT's power for deep hierarchies in Sec. 6.

Remark 5.41. *The LGT offers numerical stability through linear bounds, simpler optimization (no λ tuning), and algebraic purity by preserving graded vector space equivalence via normalization. It excels in tasks with moderate hierarchies (e.g., NLP parsing, Sec. 8), but may require additional layers for deep hierarchies compared to exponential scaling. Its linear weighting aligns with trends in efficient transformers, offering robust, interpretable solutions for structured data. Linear avoids EGT's explosion but may underfit ultra-deep hierarchies.*

6. Exponentially Graded Transformer Model

The Exponentially 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 [4]. Extending the framework of graded vector spaces from Sec. 2, [1], and Graded Neural Networks (GNNs) [2], 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, establishes its mathematical properties, and details its architecture, laying the foundation for training and applications in Sections 7 and 8. Exponential grading builds on linear from Sec. 5 for deeper hierarchies.

6.1. **Definition of Exponentially Graded Transformer.** 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 $\mathrm{M}_{\mathbf{q},\lambda}$ with grading tuple $\mathbf{q} = (q_0, \ldots, q_{d-1})$, $q_i \geq 0$, $\lambda > 1$, as defined in Eq. (33), and a standard transformer

$$\mathcal{T} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n,$$

define the map

$$\begin{aligned} \phi_{\mathbf{q},\lambda} &: (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n \\ \phi_{\mathbf{q},\lambda}(X) &= (\mathrm{M}_{\mathbf{q},\lambda}\, \mathbf{x}_i)_{i=1}^n, \end{aligned} \tag{68}$$

where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

Definition 6.1. *An **Exponentially Graded Transformer** (EGT)*

$$\bar{\Psi}_{\mathbf{q},\lambda} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n \tag{69}$$

is the map defined as

$$Y = \bar{\Psi}_{\mathbf{q},\lambda}(X) = \mathcal{T}(\phi_{\mathbf{q},\lambda}(X)). \tag{70}$$

The following is elementary and similar to the LGT case.

Lemma 6.2. *$\bar{\Psi}_{\mathbf{q},\lambda} : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$ is a well-defined map.*

Remark 6.3. *An Exponentially Graded Transformer $\bar{\Psi}_{\mathbf{q},\lambda}$ is generally non-linear, as the standard transformer $\mathcal{T}$ includes non-linear operations, such as the* **smax** *function and feed-forward layers with activation functions (e.g., ReLU). However, $\phi_{\mathbf{q},\lambda}$ is linear, since the grading map is a linear transformation applied component-wise.*

With EGT defined, we specify its key component: exponentially graded attention, extending the linear variant from Def. 5.13.

Definition 6.4 (Exponentially Graded Attention)**.** *For an input sequence $X \in \mathrm{Mat}_{n,d}(\mathbb{R})$, the **exponentially graded attention** mechanism for head i in layer l is:*

$$\mathcal{A}_{\mathbf{q}_i,\lambda}(Q_i, K_i, V_i) = \mathbf{smax}\left(\frac{Q_i\, \mathrm{M}_{\mathbf{q}_i,\lambda}\, K_i^T}{\sqrt{d_k}}\right) V_i, \tag{71}$$

where Q_i, K_i, V_i are as in Sec. 4, and

$$\mathrm{M}_{\mathbf{q}_i,\lambda} = \mathrm{diag}(\lambda^{q_{i,0}}, \ldots, \lambda^{q_{i,d_k-1}}) \in \mathrm{Mat}_{d_k,d_k}(\mathbb{R}) \tag{72}$$

*is the **head-specific grading transformation** with tuple $\mathbf{q}_i = (q_{i,0}, \ldots, q_{i,d_k-1})$, $q_{i,j} \geq 0$, $\lambda > 1$. It employs the positive definite bilinear form*

$$\langle \mathbf{q}_i, \mathbf{k}_j \rangle_{\mathbf{q}_i,\lambda} = \mathbf{q}_i^T\, \mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{k}_j = \sum_{k=0}^{d_k-1} \lambda^{q_{i,k}} q_{ik} k_{jk}. \tag{73}$$

Proposition 6.5 (Properties of Head-Specific Exponential Grading)**.** *Let $\mathrm{M}_{\mathbf{q}_i,\lambda} = \mathrm{diag}(\lambda^{q_{i,0}}, \ldots, \lambda^{q_{i,d_k-1}})$ with $\lambda > 1$, $q_{i,j} \geq 0$, and $q_{i,\max} = \max_j q_{i,j}$. Then:*

- *i)* Invertibility*: $\mathrm{M}_{\mathbf{q}_i,\lambda}$ is invertible with inverse $\mathrm{diag}(\lambda^{-q_{i,0}}, \ldots, \lambda^{-q_{i,d_k-1}})$.*
- *ii)* Scaling*: $\mathrm{M}_{\mathbf{q}_i,\lambda}(\mu\mathbf{x}) = \mu\, \mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{x}$ for $\mu \in \mathbb{R}$.*
- *iii)* Norm Bound*: $\| \mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{x} \|_2 \leq \lambda^{q_{i,\max}} \|\mathbf{x}\|_2$.*
- *iv)* Spectral Norm*: $\| \mathrm{M}_{\mathbf{q}_i,\lambda} \|_2 = \lambda^{q_{i,\max}}$.*
- *v)* Lipschitz Continuity*: The map $\mathbf{x} \mapsto \mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{x}$ has constant $\lambda^{q_{i,\max}}$.*

Proof. The proof follows analogously from the linear case in Lem. 5.5, replacing $q_{i,j}$ with $\lambda^{q_{i,j}}$. □

The assumptions $q_{i,j} \geq 0$, $\lambda > 1$ ensure key properties of the bilinear form $\langle \cdot, \cdot \rangle_{\mathbf{q}_i,\lambda}$, including positive definiteness and concentration on high-grade features.

Lemma 6.6. *The bilinear form $\langle \cdot, \cdot \rangle_{\mathbf{q}_i,\lambda}$ on $\mathbb{R}^{d_k}$ is positive definite for $\lambda > 0$.*

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

$$\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbf{q}_i,\lambda} = \mathbf{x}^T\, \mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{x} = \sum_{j=0}^{d_k-1} \lambda^{q_{i,j}} x_j^2. \tag{74}$$

Since $\lambda > 0$ and $q_{i,j} \geq 0$, $\lambda^{q_{i,j}} > 0$. The sum is positive if $\mathbf{x} \neq 0$ and zero otherwise. □

Lemma 6.7. *For $\lambda > 1$, the exponentially graded attention weights*

$$\alpha_{ij} = \mathbf{smax}\left(\frac{\mathbf{q}_i^T \,\mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{k}_j}{\sqrt{d_k}}\right)_{ij} \tag{75}$$

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

Proof. The score $s_{ij} = \sum_{k=0}^{d_k-1} \lambda^{q_{i,k}} q_{ik} k_{jk}$ is dominated by terms near $q_{i,\max}$, as $\lambda^{q_{i,k}-q_{i,\max}}$ decays exponentially. Under bounds $|q_{ik}|, |k_{jk}| \le C$,

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

with error $O(\lambda^{q_{i,k}-q_{i,\max}})$. Thus, $\alpha_{ij} \propto \exp(s_{ij}/\sqrt{d_k})$ concentrates on high-grade features. □

With the foundational properties of the bilinear form established, we now examine the stability of the attention scores under perturbations in queries and keys.

Proposition 6.8. *The attention score $s_{ij} = \mathbf{q}_i^T \,\mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{k}_j$ is Lipschitz continuous with respect to $\mathbf{q}_i, \mathbf{k}_j$, with constant at most $\lambda^{q_{i,\max}} C$, where $C > 0$ bounds $\|\mathbf{q}_i\|_2, \|\mathbf{k}_j\|_2$, and $q_{i,\max} = \max_j q_{i,j}$.*

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

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

By Prop. 6.5, $\|\,\mathrm{M}_{\mathbf{q}_i,\lambda}\,\|_2 = \lambda^{q_{i,\max}}$, so

$$\begin{aligned} \|\,\mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{k}_j\|_2 &\le \lambda^{q_{i,\max}} \|\mathbf{k}_j\|_2, \\ \|\,\mathrm{M}_{\mathbf{q}_i,\lambda}(\mathbf{k}_j - \mathbf{k}_j')\|_2 &\le \lambda^{q_{i,\max}} \|\mathbf{k}_j - \mathbf{k}_j'\|_2. \end{aligned} \tag{76}$$

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

$$|s_{ij} - s_{ij}'| \le \lambda^{q_{i,\max}} C(\|\mathbf{q}_i - \mathbf{q}_i'\|_2 + \|\mathbf{k}_j - \mathbf{k}_j'\|_2). \tag{77}$$

□

Recall the Sobolev space and universal approximator definitions from Def. 5.8 and Def. 5.9, respectively.

Theorem 6.9. *For any compact set $\Omega \subset (\mathbb{R}^d)^n$, the Exponentially Graded Transformer $\bar{\Psi}_{\mathbf{q},\lambda}$ is a universal approximator for continuous functions $f \in C(\Omega, (\mathbb{R}^d)^n)$.*

Moreover, for $f \in W^{k,2}(\Omega)$ with outputs depending primarily on features near $q_{\max} = \max_i q_i$, $\bar{\Psi}_{\mathbf{q},\lambda}$ requires fewer parameters than $\mathcal{T}$ to achieve error $\epsilon > 0$, due to reduced effective dimension via concentration, with $\mathrm{d}_{\text{eff}} = |\{i : q_i \ge q_{\max} - \delta\}|$ for $\delta > 0$.

Proof. The proof follows analogously to Thm. 5.10 for the linear case. The grading map $\phi_{\mathbf{q},\lambda}$ is invertible under the assumptions (Prop. 6.5), enabling the composition argument for universality.

Define

$$g = f \circ \phi_{\mathbf{q},\lambda}^{-1} : \phi_{\mathbf{q},\lambda}(\Omega) \to (\mathbb{R}^d)^n. \tag{78}$$

Since f and $\phi_{\mathbf{q},\lambda}^{-1}$ are continuous, $g \in C(\phi_{\mathbf{q},\lambda}(\Omega), (\mathbb{R}^d)^n)$. By [3], there exists $\mathcal{T}$ such that:

$$\sup_{Z \in \phi_{\mathbf{q},\lambda}(\Omega)} \|\mathcal{T}(Z) - g(Z)\|_2 < \epsilon. \tag{79}$$

For $Z = \phi_{\mathbf{q},\lambda}(X)$, $\bar{\Psi}_{\mathbf{q},\lambda}(X) = \mathcal{T}(Z) \approx g(Z) = f(X)$, yielding the bound.

For Sobolev $f \in W^{k,2}(\Omega)$, $\mathcal{T}$ requires $O(\epsilon^{-2/k} d)$ parameters [3]. With concentration on high-grade features near $q_{\max}$ (Lem. 6.7), low-grade contributions decay as $O(\lambda^{q_i - q_{\max}})$, yielding effective effective complexity $O(\epsilon^{-2/k}\, \mathrm{d}_{\text{eff}})$, as in [1].

The EGT attention (Prop. 6.10) prioritizes high grades, with reduction proportional to the grading concentration. □

Let $\lambda > 1$ and $q_{i,j} \ge 0$. Let $\mathcal{A}_{\mathbf{q}_i,\lambda}(Q_i, K_i, V_i)$ denote the exponentially graded attention output and $\mathcal{A}(Q_i, K_i, V_i)$ the standard attention, as in Def. 6.4.

Grading modulates attention rank, enabling adaptive expressivity for hierarchical features.

Proposition 6.10. *The singular values of $Q_i \,\mathrm{M}_{\mathbf{q}_i,\lambda}\, K_i^T$ are scaled by at most $\lambda^{q_{i,\max}}$ compared to those of $Q_i K_i^T$, where $q_{i,\max} = \max_j q_{i,j}$. Consequently, the effective numerical rank of $\mathcal{A}_{\mathbf{q}_i,\lambda}(Q_i, K_i, V_i)$ may increase or decrease depending on the singular value distribution of $Q_i K_i^T$ and $\mathbf{q}_i$.*

Proof. Let $Q_i K_i^T = U\Sigma V^T$ with singular values $\sigma_j > 0$. Since $\| \mathrm{M}_{\mathbf{q}_i,\lambda} \|_2 = \lambda^{q_{i,\max}}$ (Prop. 6.5),

$$\|Q_i \,\mathrm{M}_{\mathbf{q}_i,\lambda}\, K_i^T\|_2 \le \lambda^{q_{i,\max}} \|Q_i K_i^T\|_2. \tag{80}$$

Thus, $\tilde{\sigma}_j \le \lambda^{q_{i,\max}} \sigma_j$. Dimensions near $q_{i,\max}$ dominate, suppressed by $\lambda^{q_{i,j}-q_{i,\max}}$ otherwise (Lem. 6.7), adapting rank accordingly. □

We now demonstrate the expressivity of exponentially graded attention, which can approximate arbitrary non-negative matrices, analogous to the linear case in Prop. 5.14. Recall that the Frobenius norm is $\|A\|_F = \sqrt{\mathrm{tr}(A^T A)}$, where tr denotes the trace.

Proposition 6.11. *For any row-stochastic matrix $A_0 \in \mathrm{Mat}_{n,d_k}(\mathbb{R})$ with non-negative entries, there exist $Q, K, V \in \mathrm{Mat}_{n,d_k}(\mathbb{R})$, $\mathbf{q}_i$, and $\lambda > 1$ such that $\mathcal{A}_{\mathbf{q}_i,\lambda}(Q, K, V)$ approximates A_0 with*

$$\|\mathcal{A}_{\mathbf{q}_i,\lambda}(Q, K, V) - A_0\|_F \le \delta. \tag{81}$$

Proof. Let A_0 have non-negative entries summing to 1 per row. Construct A_0' by replacing near-zeros with $\epsilon = \delta/(2\sqrt{nd_k})$ and renormalizing. Choose Q, K such that

$$QK^T = \sqrt{d_k} \log A_0', \tag{82}$$

ensuring $\mathbf{smax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \approx A_0'$. Set $V = I$, yielding standard attention close to A_0', with error $\le \delta/2$. For graded, set $q_{i,j} = cj$ small $c > 0$, large λ; by Lem. 6.7, $\mathrm{M}_{\mathbf{q}_i,\lambda} \approx I$, so error $\le \delta/2$. Total: $\le \delta$. □

Recall the VC dimension definition from Def. 5.15.

Let $\mathcal{F}$ be the hypothesis class of the Exponentially Graded Transformer $\bar{\Psi}_{\mathbf{q},\lambda} = \mathcal{T} \circ \phi_{\mathbf{q},\lambda}$.

Grading reduces effective complexity, as quantified by VC dimension, lowering sample requirements.

Proposition 6.12. *The VC dimension satisfies*

$$\mathrm{vc}(\mathcal{F}) \le c \cdot N \cdot h \cdot d \log(Nhd), \tag{83}$$

reduced from the standard transformer's

$$\mathrm{vc}(\mathcal{F}_{\mathcal{T}}) \le c \cdot Nhd^2 \log(Nhd), \tag{84}$$

where c is a constant and N is the number of layers. The reduction is proportional to the concentration factor from grading (Lem. 6.7), lowering sample complexity $O(\mathrm{vc}(\mathcal{F}) \log(1/\epsilon)/\epsilon^2 + \log(1/\delta))$.

Proof. Analogous to Prop. 5.16, the effective parameter count is $O(Nhd)$ via exponential concentration on high-grade features, yielding the bound and reduction. □

Grading enhances robustness to perturbations, as we now quantify for EGT.

Proposition 6.13. *For noise $\Delta \in (\mathbb{R}^d)^n$ with $\|\Delta\|_2 \le \epsilon$, the output error satisfies*

$$\| \bar{\Psi}_{\mathbf{q},\lambda}(X + \Delta) - \bar{\Psi}_{\mathbf{q},\lambda}(X)\|_2 \le \Lambda\epsilon, \tag{85}$$

where $\Lambda \le L_{\mathcal{T}} (\lambda^{\mathbf{q}_{\max}})^{Nh+3}$, $L_{\mathcal{T}}$ is the Lipschitz constant of $\mathcal{T}$, and $q_{\max} = \max_i q_i$.

Proof. Analogous to Prop. 5.12, the perturbation propagates as

$$\|\phi_{\mathbf{q},\lambda}(X + \Delta) - \phi_{\mathbf{q},\lambda}(X)\|_2 \le \lambda^{\mathbf{q}_{\max}} \epsilon$$

by the norm bound in Prop. 6.5. Attention adds per-head factors up to $\lambda^{q_{i,\max}} \le \lambda^{\mathbf{q}_{\max}}$ (Prop. 6.8). With N layers and h heads, plus input/feed-forward/output, the overall bound follows, ensuring robustness for noisy data like genomics or flows. □

We conclude the properties with computational complexity, confirming grading adds no asymptotic overhead.

Proposition 6.14. *The exponentially graded attention $\mathcal{A}_{\mathbf{q}_i,\lambda}(Q, K, V)$ has complexity $O(n^2 d_k)$, matching the complexity of standard attention.*

Proof. Analogous to Prop. 5.18, the diagonal $\mathrm{M}_{\mathbf{q}_i,\lambda}$ adds $O(nd_k)$ for scaling, absorbed into matrix multiplications QMK^T and softmax-V ($O(n^2 d_k)$ each). Thus, total $O(n^2 d_k)$. □

6.2. **Architecture.** The Exponentially Graded Transformer embeds grading transformations $\mathrm{L}_{\mathbf{q},\lambda}$ across its components to enhance feature prioritization for structured sequence data. Below, we detail the input representation, positional encoding, attention variants, feed-forward layers, and output layer, ensuring consistent hierarchical emphasis.

For each token $\mathbf{x}_i \in \mathbb{R}^d$ in the input sequence $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, the exponentially graded input representation is:

$$\mathbf{x}_i' = \mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{x}_i, \quad \mathbf{x}_i'' = \frac{\mathbf{x}_i'}{\|\mathbf{x}_i'\|_2}, \quad \mathbf{h}_i = \sigma(W\mathbf{x}_i'' + \mathbf{b}),$$

where $\mathbf{q}_i = (q_{i,0}, \ldots, q_{i,d-1}) \in \mathbb{R}^d_{\geq 0}$, and $\lambda > 1$, $W \in \mathbb{R}^{m\times d}$, $\mathbf{b} \in \mathbb{R}^m$, and σ is typically ReLU. Here, $\mathrm{M}_{\mathbf{q}_i,\lambda} = \mathrm{diag}(\lambda^{q_{i,0}}, \ldots, \lambda^{q_{i,d-1}})$ is the exponential grading matrix associated with token i.

Normalization prevents numerical instability due to large λ^{q_i}. Positional encodings capture sequence order, enhanced with grading to prioritize certain positions (e.g., earlier tokens in hierarchical tasks like parsing). Standard positional encodings are

$$\mathrm{Pe}(\mathbf{t}, 2i) = \sin\left(\frac{\mathbf{t}}{10000^{2i/d}}\right),$$
$$\mathrm{Pe}(\mathbf{t}, 2i+1) = \cos\left(\frac{\mathbf{t}}{10000^{2i/d}}\right),$$

for position $\mathbf{t}$ and dimension i. Exponentially graded encodings are

$$\mathrm{Pe}'(\mathbf{t}, i) = \lambda^{q_\mathbf{t}}\, \mathrm{Pe}(\mathbf{t}, i), \quad q_\mathbf{t} = f(\mathbf{t}),$$

where $f(\mathbf{t}) = -\alpha\mathbf{t}$, $\alpha > 0$, emphasizes earlier positions. The input to the attention mechanism is

$$\mathbf{z}_i = \mathbf{x}_i'' + \mathrm{Pe}'(\mathbf{t}_i, \cdot), \quad \mathbf{z}_i' = \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2},$$

and attention scores are computed as

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

where $W_Q, W_K \in \mathbb{R}^{d\times d_k}$. The exponentially graded attention mechanism (Def. 6.4) prioritizes high-grade features in attention scores. Alternative variants include:

Exponentially Graded Scores:

$$\begin{aligned} \mathbf{sc}_{ij} &= \sum_{k=1}^{d_k} \lambda^{q_{i,k}} q_{ik} k_{jk}, \\ \mathrm{M}_{\mathbf{q}_i,\lambda}{}^K &= \mathrm{diag}(\lambda^{q_{i,1}}, \ldots, \lambda^{q_{i,d_k}}), \\ \mathbf{sc}_{ij} &= (Q\, \mathrm{M}_{\mathbf{q}_i,\lambda}{}^K\, K^T)_{ij}, \end{aligned} \tag{86}$$

Exponentially Graded Queries/Keys:

$$Q' = \mathrm{M}_{\mathbf{q}_i,\lambda}\, Q, \quad K' = \mathrm{M}_{\mathbf{q}_i,\lambda}\, K, \quad \mathbf{sc}_{ij} = \mathbf{q}_i^T\, \mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{k}_j \tag{87}$$

Exponentially Graded Multi-Head:

$$\mathrm{Head}_h = \mathbf{smax}\left(\frac{(M_{\mathbf{q}_h,\lambda} Q_h)(M_{\mathbf{q}_h,\lambda} K_h)^T}{\sqrt{d_k}}\right) V_h, \tag{88}$$

with distinct $\mathbf{q}_h = (q_{h,0}, \ldots, q_{h,d_k-1})$ per head.

Exponentially Graded Values:

$$\begin{aligned} V' &= \mathrm{M}_{\mathbf{q}_i,\lambda}\, V, \\ \mathbf{o}_i &= \sum_{j=1}^{n} \alpha_{ij} (\mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathbf{v}_j), \\ \alpha_{ij} &= \mathbf{smax}\left(\frac{\mathbf{q}_i^T \mathbf{k}_j}{\sqrt{d_k}}\right). \end{aligned} \tag{89}$$

Definition 6.15 (Feed-Forward and Output Layers)**.** *The* ***exponentially graded feed-forward network*** *is:*

$$\mathrm{Fnn}'(\mathbf{x}) = \mathrm{M}_{\mathbf{q}_i,\lambda}\, \mathrm{Fnn}(\mathbf{x}), \quad \mathbf{h}' = \frac{\mathrm{Fnn}'(\mathbf{x})}{\|\,\mathrm{Fnn}'(\mathbf{x})\|_2}, \tag{90}$$

where $\mathrm{Fnn}(\mathbf{x}) = \mathrm{ReLU}(\mathbf{x}W_1+\mathbf{b}_1)W_2+\mathbf{b}_2$, $W_1 \in \mathbb{R}^{d\times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff}\times d}$, *and* d_{ff} *is the hidden dimension. The* ***exponentially graded output*** *is:*

$$\mathbf{h}' = \mathrm{M}_{\mathbf{q}_i,\lambda}\,\mathbf{h}, \quad \mathbf{z} = W_{out}\mathbf{h}' + \mathbf{b}_{out}, \tag{91}$$

followed by $\mathbf{smax}(\mathbf{z})$.

Remark 6.16. *Here, the same grading matrix* $\mathrm{M}_{\mathbf{q}_i,\lambda}$ *associated with token* i *is applied in both the feed-forward and output layers, ensuring consistent feature scaling throughout the architecture.*

Theorem 6.17. *The mapping* $\mathbf{x}_i \mapsto \mathbf{x}_i''$ *is Lipschitz continuous with constant at most* $\lambda^{\mathbf{q}_{\max}}$, *where* $\mathbf{q}_{\max} = \max_j\{q_{i,j}\}$, *for fixed token* i, *assuming* $\lambda > 1$ *and* $q_i \geq 0$.

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

$$\mathbf{x}' = \mathrm{M}_{\mathbf{q}_i,\lambda}\,\mathbf{x}, \quad \mathbf{y}' = \mathrm{M}_{\mathbf{q}_i,\lambda}\,\mathbf{y}.$$

From Prop. 6.5, $\|\,\mathrm{M}_{\mathbf{q}_i,\lambda}(\mathbf{x}-\mathbf{y})\|_2 \leq \lambda^{\mathbf{q}_{\max}}\|\mathbf{x}-\mathbf{y}\|_2$. Thus:

$$\|\mathbf{x}'-\mathbf{y}'\|_2 = \|\,\mathrm{M}_{\mathbf{q}_i,\lambda}(\mathbf{x}-\mathbf{y})\|_2 \leq \lambda^{\mathbf{q}_{\max}}\|\mathbf{x}-\mathbf{y}\|_2.$$

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:

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

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

Corollary 6.18 (Bounded Activations)**.** *For all* i, *if* $\mathbf{x}_i' \neq 0$, *then* $\|\mathbf{x}_i''\|_2 = 1$. *That is, the normalized representations lie on the unit sphere in* $\mathbb{R}^d$.

Lemma 6.19. *Let* $\mathbf{q}_i = (q_{i,0},\ldots,q_{i,d-1}) \in \mathbb{R}^d_{\geq 0}$. *Then the Jacobian of the mapping* $\mathbf{x}_i \mapsto \mathbf{x}_i' = \mathrm{M}_{\mathbf{q}_i,\lambda}\,\mathbf{x}_i$ *has operator norm* $\lambda^{q_{\max}}$, *where* $q_{\max} = \max_j q_{i,j}$.

Proof. Since $\mathrm{M}_{\mathbf{q}_i,\lambda}$ is diagonal with entries $\lambda^{q_{i,j}}$, its operator norm (i.e., largest singular value) is

$$\|\,\mathrm{M}_{\mathbf{q}_i,\lambda}\,\|_2 = \max_j \lambda^{q_{i,j}} = \lambda^{q_{\max}}.$$

□

Proposition 6.20 (Exponential Positional Bias)**.** *Let* $\mathbf{x}_i'' \in \mathbb{R}^d$ *be a unit vector, and let* $\mathrm{Pe}(\mathbf{t}) \in \mathbb{R}^d$ *be fixed positional encodings with* $\|\,\mathrm{Pe}(\mathbf{t})\|_2 = c$ *for all* $\mathbf{t}$. *Define exponentially graded encodings*

$$\mathrm{Pe}'(\mathbf{t},i) := \lambda^{-\alpha\mathbf{t}}\,\mathrm{Pe}(\mathbf{t},i), \quad \lambda > 1,\ \alpha > 0,$$

and define

$$\mathbf{z}_i := \mathbf{x}_i'' + \mathrm{Pe}'(\mathbf{t}_i), \quad \mathbf{sc}_{ij} := \langle \mathbf{z}_i, \mathbf{z}_j\rangle.$$

Then, for fixed $\mathbf{x}_i''$, *if* $\mathbf{t}_j < \mathbf{t}_k$, *we have:*

$$\mathbf{sc}_{ij} > \mathbf{sc}_{ik}.$$

That is, attention scores are biased toward earlier positions.

Proof. We compute the unnormalized attention scores:

$$\mathbf{sc}_{ij} = \langle \mathbf{z}_i, \mathbf{z}_j\rangle = \langle \mathbf{x}_i'' + \mathrm{Pe}'(\mathbf{t}_i), \mathbf{x}_j'' + \mathrm{Pe}'(\mathbf{t}_j)\rangle.$$

Expanding this:

$$\mathbf{sc}_{ij} = \langle \mathbf{x}_i'', \mathbf{x}_j''\rangle + \langle \mathbf{x}_i'', \mathrm{Pe}'(\mathbf{t}_j)\rangle + \langle \mathrm{Pe}'(\mathbf{t}_i), \mathbf{x}_j''\rangle + \langle \mathrm{Pe}'(\mathbf{t}_i), \mathrm{Pe}'(\mathbf{t}_j)\rangle.$$

Substitute $\mathrm{Pe}'(\mathbf{t}) = \lambda^{-\alpha\mathbf{t}}\,\mathrm{Pe}(\mathbf{t})$ and define:

$$A := \langle \mathbf{x}_i'', \mathbf{x}_j''\rangle, \quad B_j := \langle \mathbf{x}_i'', \mathrm{Pe}(\mathbf{t}_j)\rangle, \quad C_j := \langle \mathrm{Pe}(\mathbf{t}_i), \mathrm{Pe}(\mathbf{t}_j)\rangle.$$

Then:

$$\mathbf{sc}_{ij} = A + \lambda^{-\alpha\mathbf{t}_j}B_j + \lambda^{-\alpha\mathbf{t}_i}\langle \mathrm{Pe}(\mathbf{t}_i), \mathbf{x}_j''\rangle + \lambda^{-\alpha(\mathbf{t}_i+\mathbf{t}_j)}C_j.$$

Now fix all terms except $\mathbf{t}_j$. Since $\lambda > 1$ and $\alpha > 0$, the scalars $\lambda^{-\alpha \mathbf{t}_j}$ and $\lambda^{-\alpha(\mathbf{t}_i + \mathbf{t}_j)}$ are strictly decreasing in $\mathbf{t}_j$. Thus, under the assumption that B_j and C_j are non-negative or bounded below, we conclude that:

$$\mathbf{t}_j < \mathbf{t}_k \quad \Longrightarrow \quad \mathbf{sc}_{ij} > \mathbf{sc}_{ik}.$$

This proves that the attention score $\mathbf{sc}_{ij}$ is biased toward earlier positions. □

Lemma 6.21 (Lipschitz Stability in Position)**.** *Let* $\mathbf{z}_i(\mathbf{t}) := \mathbf{x}_i'' + \mathrm{Pe}'(\mathbf{t})$*, where* $\mathrm{Pe}'(\mathbf{t}) := \lambda^{f(\mathbf{t})} \mathrm{Pe}(\mathbf{t})$*, and assume* $\|\mathrm{Pe}(\mathbf{t})\|_2 \leq c$ *for all* $\mathbf{t} \in \mathbb{N}$*. Then the mapping*

$$\mathbf{t} \mapsto \mathbf{z}_i'(\mathbf{t}) := \frac{\mathbf{z}_i(\mathbf{t})}{\|\mathbf{z}_i(\mathbf{t})\|_2}$$

is Lipschitz continuous with constant bounded by $C\lambda^{\mathbf{q}_{\max}}$*, where* $\mathbf{q}_{\max} = \sup_{\mathbf{t}} |f(\mathbf{t})|$*, and* C *depends on* α*,* $\ln \lambda$*, and* c*.*

Proof. Let $\mathbf{t}, \mathbf{t}' \in \mathbb{N}$, and consider:

$$\mathbf{z}_i(\mathbf{t}) = \mathbf{x}_i'' + \lambda^{f(\mathbf{t})} \mathrm{Pe}(\mathbf{t}), \quad \mathbf{z}_i(\mathbf{t}') = \mathbf{x}_i'' + \lambda^{f(\mathbf{t}')} \mathrm{Pe}(\mathbf{t}').$$

Then:

$$\|\mathbf{z}_i(\mathbf{t}) - \mathbf{z}_i(\mathbf{t}')\|_2 = \|\lambda^{f(\mathbf{t})} \mathrm{Pe}(\mathbf{t}) - \lambda^{f(\mathbf{t}')} \mathrm{Pe}(\mathbf{t}')\|_2.$$

Add and subtract $\lambda^{f(\mathbf{t}')} \mathrm{Pe}(\mathbf{t})$:

$$= \|(\lambda^{f(\mathbf{t})} - \lambda^{f(\mathbf{t}')}) \mathrm{Pe}(\mathbf{t}) + \lambda^{f(\mathbf{t}')}(\mathrm{Pe}(\mathbf{t}) - \mathrm{Pe}(\mathbf{t}'))\|_2.$$

Using triangle inequality and $\|\mathrm{Pe}(\mathbf{t})\|_2 \leq c$:

$$\leq |\lambda^{f(\mathbf{t})} - \lambda^{f(\mathbf{t}')}| \cdot c + \lambda^{f(\mathbf{t}')} \cdot \|\mathrm{Pe}(\mathbf{t}) - \mathrm{Pe}(\mathbf{t}')\|_2.$$

Now assume $f(\mathbf{t}) = -\alpha \mathbf{t}$. The map $t \mapsto \lambda^{-\alpha t}$ is smooth, and by the mean value theorem:

$$|\lambda^{f(\mathbf{t})} - \lambda^{f(\mathbf{t}')}| = |\lambda^{-\alpha \mathbf{t}} - \lambda^{-\alpha \mathbf{t}'}| \leq \alpha \ln \lambda \cdot \lambda^{\alpha |\mathbf{t} - \mathbf{t}'|} \cdot |\mathbf{t} - \mathbf{t}'|.$$

So we have

$$\|\mathbf{z}_i(\mathbf{t}) - \mathbf{z}_i(\mathbf{t}')\|_2 \leq C_1 \lambda^{|f(\mathbf{t})|} |\mathbf{t} - \mathbf{t}'|,$$

where C_1 depends on α, $\ln \lambda$, and c, and assuming $\|\mathrm{Pe}(\mathbf{t}) - \mathrm{Pe}(\mathbf{t}')\|_2 \leq L|\mathbf{t} - \mathbf{t}'|$ for some L (true for sinusoidal encodings).

Since normalization $\mathbf{z} \mapsto \mathbf{z}/\|\mathbf{z}\|_2$ is 1-Lipschitz on $\mathbb{R}^d \setminus \{0\}$, we conclude:

$$\|\mathbf{z}_i'(\mathbf{t}) - \mathbf{z}_i'(\mathbf{t}')\|_2 \leq \frac{\|\mathbf{z}_i(\mathbf{t}) - \mathbf{z}_i(\mathbf{t}')\|_2}{\min(\|\mathbf{z}_i(\mathbf{t})\|_2, \|\mathbf{z}_i(\mathbf{t}')\|_2)}.$$

Assuming $\|\mathbf{z}_i(\mathbf{t})\|_2 \geq \epsilon > 0$, we obtain:

$$\|\mathbf{z}_i'(\mathbf{t}) - \mathbf{z}_i'(\mathbf{t}')\|_2 \leq \frac{C_1 \lambda^{\mathbf{q}_{\max}}}{\epsilon} |\mathbf{t} - \mathbf{t}'| = C\lambda^{\mathbf{q}_{\max}} |\mathbf{t} - \mathbf{t}'|,$$

where $\mathbf{q}_{\max} := \sup_{\mathbf{t}} |f(\mathbf{t})|$. This proves Lipschitz continuity. □

Theorem 6.22 (Lipschitz Stability of Graded Attention Scores)**.** *Fix a grading vector* $\mathbf{q}_i = (q_{i,1}, \ldots, q_{i,d_k}) \in \mathbb{R}^{d_k}_{\geq 0}$*, and let* $\mathrm{M}_{\mathbf{q}_i,\lambda} := \mathrm{diag}(\lambda^{q_{i,1}}, \ldots, \lambda^{q_{i,d_k}})$*. Then the map*

$$(\mathbf{q}, \mathbf{k}) \mapsto \mathbf{sc}_{ij} := \mathbf{q}^T \mathrm{M}_{\mathbf{q}_i,\lambda} \mathbf{k}$$

is Lipschitz continuous on bounded subsets of $\mathbb{R}^{d_k} \times \mathbb{R}^{d_k}$*, with*

$$\Lambda(\mathbf{sc}_{ij}) \leq \lambda^{q_{i,\max}} C,$$

where $q_{i,\max} := \max_j q_{i,j}$*, and* C *depends on bounds on* $\|\mathbf{q}\|_2$*,* $\|\mathbf{k}\|_2$*.*

Proof. Let $\mathbf{q}, \mathbf{q}' \in \mathbb{R}^{d_k}$, $\mathbf{k}, \mathbf{k}' \in \mathbb{R}^{d_k}$. Then

$$\begin{aligned}
&\left|\mathbf{q}^T \mathrm{M}_{\mathbf{q}_i,\lambda} \mathbf{k} - \mathbf{q}'^T \mathrm{M}_{\mathbf{q}_i,\lambda} \mathbf{k}'\right| = \\
&= \left|\mathbf{q}^T \mathrm{M}_{\mathbf{q}_i,\lambda} \mathbf{k} - \mathbf{q}'^T \mathrm{M}_{\mathbf{q}_i,\lambda} \mathbf{k} + \mathbf{q}'^T \mathrm{M}_{\mathbf{q}_i,\lambda} \mathbf{k} - \mathbf{q}'^T \mathrm{M}_{\mathbf{q}_i,\lambda} \mathbf{k}'\right| \\
&\leq \|\mathbf{q} - \mathbf{q}'\|_2 \cdot \|\mathrm{M}_{\mathbf{q}_i,\lambda} \mathbf{k}\|_2 + \|\mathbf{q}'\|_2 \cdot \|\mathrm{M}_{\mathbf{q}_i,\lambda}(\mathbf{k} - \mathbf{k}')\|_2 \\
&\leq \lambda^{q_{i,\max}} \left(\|\mathbf{q} - \mathbf{q}'\|_2 \cdot \|\mathbf{k}\|_2 + \|\mathbf{q}'\|_2 \cdot \|\mathbf{k} - \mathbf{k}'\|_2\right) \\
&\leq \lambda^{q_{i,\max}} C \left(\|\mathbf{q} - \mathbf{q}'\|_2 + \|\mathbf{k} - \mathbf{k}'\|_2\right),
\end{aligned}$$

where the last inequality assumes $\|\mathbf{q}\|_2, \|\mathbf{q}'\|_2, \|\mathbf{k}\|_2, \|\mathbf{k}'\|_2 \leq C$. Thus, $\Lambda(\mathbf{sc}_{ij}) \leq \lambda^{q_{i,\max}} C$. □

Proposition 6.23. *Distinct grading tuples* $\mathbf{q}_h$ *in the Exponentially Graded Multi-Head variant enhance representational capacity.*

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

$$\text{Head}_h = \mathbf{smax}\left(\frac{(\text{M}_{\mathbf{q}_i,\lambda}\, Q_h)(\text{M}_{\mathbf{q}_i,\lambda}\, K_h)^T}{\sqrt{d_k}}\right) V_h,$$

where $\text{M}_{\mathbf{q}_i,\lambda} = \text{diag}(\lambda^{q_{h,0}}, \ldots, \lambda^{q_{h,d_k-1}})$. Distinct grading tuples $\mathbf{q}_h = (q_{h,0}, \ldots, q_{h,d_k-1})$ produce distinct scaling across dimensions. This induces different projections of the queries and keys across heads, effectively spanning different graded subspaces. The resulting set of heads captures a more diverse range of dependencies. Since each $\text{M}_{\mathbf{q}_i,\lambda}$ alters the singular spectrum of the scaled product $Q_h K_h^T$, the overall concatenated representation spans a richer subspace of $\mathbb{R}^d$, improving expressivity. □

Proposition 6.24. *The exponentially graded feed-forward mapping* $\mathbf{x} \mapsto \text{Fnn}'(\mathbf{x})$ *has Lipschitz constant at most* $\lambda^{\mathbf{q}_{\max}} L_{\text{Fnn}}$.

Proof. Let $\text{Fnn}'(\mathbf{x}) = \text{M}_{\mathbf{q}_i,\lambda}\, \text{Fnn}(\mathbf{x})$. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$\begin{aligned} \|\,\text{Fnn}'(\mathbf{x}) - \text{Fnn}'(\mathbf{y})\|_2 &= \|\,\text{M}_{\mathbf{q}_i,\lambda}(\text{Fnn}(\mathbf{x}) - \text{Fnn}(\mathbf{y}))\|_2 \\ &\le \lambda^{\mathbf{q}_{\max}} \|\,\text{Fnn}(\mathbf{x}) - \text{Fnn}(\mathbf{y})\|_2 \\ &\le \lambda^{\mathbf{q}_{\max}} L_{\text{Fnn}} \|\mathbf{x} - \mathbf{y}\|_2. \end{aligned}$$

Normalization is 1-Lipschitz, so the overall Lipschitz constant remains $\lambda^{\mathbf{q}_{\max}} L_{\text{Fnn}}$. □

Proposition 6.25. *The output mapping* $\mathbf{h} \mapsto \mathbf{z}$ *is Lipschitz with constant at most* $\lambda^{\mathbf{q}_{\max}} L_{out}$.

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

$$\mathbf{z}_1 = W_{\text{out}}(\text{M}_{\mathbf{q}_i,\lambda}\, \mathbf{h}_1) + \mathbf{b}_{\text{out}}, \quad \mathbf{z}_2 = W_{\text{out}}(\text{M}_{\mathbf{q}_i,\lambda}\, \mathbf{h}_2) + \mathbf{b}_{\text{out}}.$$

Then,

$$\begin{aligned} \|\mathbf{z}_1 - \mathbf{z}_2\|_2 &= \|W_{\text{out}}\, \text{M}_{\mathbf{q}_i,\lambda}(\mathbf{h}_1 - \mathbf{h}_2)\|_2 \\ &\le \|W_{\text{out}}\|_2 \cdot \|\,\text{M}_{\mathbf{q}_i,\lambda}(\mathbf{h}_1 - \mathbf{h}_2)\|_2 \\ &\le \lambda^{\mathbf{q}_{\max}} \|W_{\text{out}}\|_2 \cdot \|\mathbf{h}_1 - \mathbf{h}_2\|_2. \end{aligned}$$

Let $L_{\text{out}} = \|W_{\text{out}}\|_2$. Then $\|\mathbf{z}_1 - \mathbf{z}_2\|_2 \le \lambda^{\mathbf{q}_{\max}} L_{\text{out}} \|\mathbf{h}_1 - \mathbf{h}_2\|_2$. □

Proposition 6.26. *The Exponentially Graded Transformer has the same asymptotic complexity as the standard transformer,* $O(n^2 d + nd^2)$.

Proof. The standard transformer's complexity is dominated by attention ($O(n^2 d)$) and feed-forward layers ($O(nd^2)$). Each grading transformation $\text{M}_{\mathbf{q}_i,\lambda}$ is a diagonal matrix multiplication, costing $O(d)$ or $O(d_k)$ per token. Across n tokens, this totals $O(nd)$ per layer or component. Since $nd \ll n^2 d, nd^2$, the overall complexity remains $O(n^2 d + nd^2)$. □

Remark 6.27 (Interpretability and Applications)**.** *The Exponentially Graded Transformer efficiently models hierarchical data by prioritizing high-grade features through attention concentration (Lem. 6.7) and enhanced expressivity (Prop. 6.11). Its reduced sample complexity (Prop. 5.16) and robustness (Prop. 6.13) make it ideal for structured tasks such as algebraic geometry and genomics. The assumptions* $q_i, q_{i,j} \ge 0$, $\lambda > 1$ *ensure positive scaling, though arbitrary* q_i *may be accommodated via bounds on* $\max_i |\lambda^{q_i}|$.

7. Training and Optimization

The Exponentially Graded Transformer introduces grading transformations to prioritize hierarchical features, necessitating tailored training strategies to optimize its parameters effectively. Unlike our previous work [1,2], where grades were fixed weights of the vector space, here we treat the grading tuples $\mathbf{q} = (q_1, \ldots, q_d)$ and $\mathbf{q}_i = (q_{i,1}, \ldots, q_{i,d_k})$ as learnable parameters, optimized via gradient descent to adaptively prioritize features based on data. This shift enhances the model's flexibility but introduces challenges in numerical stability and optimization complexity. This section formalizes the exponentially graded loss function, develops regularization and optimization techniques, establishes convergence and stability guarantees, and outlines EGT-specific training considerations for structured tasks in domains like algebraic geometry and natural language processing [1,2].

7.1. **Exponentially Graded Loss Function.** The exponentially graded loss function embeds hierarchical priors by weighting prediction errors according to their grades. This ensures that structurally important features—such as high-degree polynomial terms or syntactic heads—contribute more significantly to the optimization objective.

Definition 7.1. *Let $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_m) \in (\mathbb{R}^d)^m$ denote a sequence of target outputs, and let $\hat{y}_{i,k}, y_{i,k} \in \mathbb{R}$ be the predicted and true values, respectively, for the k-th dimension of the i-th token. Let $\mathbf{q} = \mathbf{q} = (q_0, \ldots, q_{d-1}) \in \mathbb{R}^d_{\geq 0}$ be a grading vector and $\lambda > 1$ a fixed scaling factor. Then the* ***exponentially graded loss function*** *is:*

$$\mathcal{L} = \sum_{i=1}^{m} \sum_{k=0}^{d-1} \lambda^{q_k} \, \ell(\hat{y}_{i,k}, y_{i,k}),$$

where:

i) $\ell(\hat{y}_{i,k}, y_{i,k})$ is a base loss function (e.g., cross-entropy or squared error),
ii) m is the sequence length and d the output dimension,
iii) λ^{q_k} weights loss according to the grade of feature k, amplifying contributions from high-grade dimensions.

Example 7.2 (Exponentially Graded Loss in Polynomial Degree Prediction)**.** *Consider a symbolic regression task in algebraic geometry, where the Exponentially Graded Transformer predicts polynomial terms needed to compute zeta functions of hypersurfaces [9]. Let each output vector $\mathbf{y}_i \in \mathbb{R}^4$ represent predicted coefficients for monomials of degree 0 through 3, with $d = 4$. Let the grading vector be $\mathbf{q} = (0, 0.5, 1, 2)$, and set $\lambda = 2$ to prioritize higher-degree terms. The graded loss becomes:*

$$\mathcal{L} = \sum_{i=1}^{m} \left[\ell(\hat{y}_{i,0}, y_{i,0}) + 2^{0.5} \ell(\hat{y}_{i,1}, y_{i,1}) + 2\ell(\hat{y}_{i,2}, y_{i,2}) + 4\ell(\hat{y}_{i,3}, y_{i,3}) \right].$$

For instance, let $m = 2$ and assume true outputs $y_{1,3} = y_{2,3} = 1$, and predicted values $\hat{y}_{1,3} = 0.8$, $\hat{y}_{2,3} = 0.9$. Using cross-entropy loss,

$$\ell(\hat{y}, y) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}),$$

the loss contributions from the highest-degree term (degree 3) are weighted by $\lambda^2 = 4$, ensuring they dominate the total loss. This graded emphasis reduces sample complexity by guiding the model to prioritize structurally important outputs (Prop. 5.16).

7.2. **Regularization and Optimization.** The Exponentially Graded Transformer's exponential grading introduces sensitivity into the optimization process, requiring explicit regularization to control the magnitude of learned grades and specialized strategies to stabilize training. These techniques are particularly important given the multi-head structure of EGT and the exponential dependence on the grading parameters.

When the grade vectors $\mathbf{q} = (q_0, \ldots, q_{d-1})$ and $\mathbf{q}_i = (q_{i,0}, \ldots, q_{i,d_k-1})$ are treated as learnable parameters, we introduce regularization terms to prevent uncontrolled growth of the grades. The total loss becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \gamma \|\mathbf{q}\|_2^2 + \gamma' \sum_{i=1}^{h} \|\mathbf{q}_i\|_2^2,$$

where $\gamma, \gamma' > 0$ are regularization weights, and h is the number of attention heads.

Optimization proceeds using standard gradient-based methods such as Adam [12]. However, differentiating through the exponential grading map introduces additional sensitivity due to the nonlinearity of the map $q_k \mapsto \lambda^{q_k}$. For instance, in the Exponentially Graded Queries/Keys variant (cf. Def. 6.4), if the attention score is:

$$\text{Score}_{ij} = \sum_{k=0}^{d_k-1} q_{i,k} \lambda^{q_{i,k}} k_{j,k},$$

then the partial derivative with respect to a learned grade $q_{i,k}$ is:

$$\frac{\partial \text{Score}_{ij}}{\partial q_{i,k}} = (\lambda^{q_{i,k}} + q_{i,k} \lambda^{q_{i,k}} \ln \lambda) \, k_{j,k}.$$

This expression reveals the exponential sensitivity of gradients to $q_{i,k}$, especially for large values of λ or $q_{i,k}$, and motivates several stabilization techniques.

To train the Exponentially Graded Transformer effectively, we propose the following strategies:

i) **Gradient Clipping**: Prevent gradient explosion by capping the gradient norm to a threshold $\tau > 0$. For a gradient vector $\mathbf{g}$, we apply:

$$\mathbf{g}' = \min\left(1, \frac{\tau}{\|\mathbf{g}\|_2}\right)\mathbf{g},$$

when $\|\mathbf{g}\|_2 > \tau$, leaving it unchanged otherwise.

ii) **Warm-Starting**: Initialize $\mathbf{q}$ and $\mathbf{q}_i$ using domain-informed priors (e.g., for polynomial degrees, use $q_k = c \cdot k$, with $c > 0$), to bias the model toward known hierarchical structures.

iii) **Grade Annealing**: To avoid instability at early training stages, gradually increase the grading scale factor λ using a schedule:

$$\lambda_t = 1 + (\lambda_{\max} - 1) \cdot \frac{t}{T},$$

where t is the current training step and T is the total number of training steps. This λ_t is then substituted into all exponential maps λ^{q_k}, ensuring consistent scaling.

iv) **Multi-Head Grade Coordination**: To prevent diverging priorities among heads, we introduce a coordination regularizer:

$$\mathcal{L}_{\text{coord}} = \sum_{i=1}^{h} \left\| \mathbf{q}_i - \frac{1}{h}\sum_{j=1}^{h} \mathbf{q}_j \right\|_2^2,$$

which is added to the total loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \gamma\|\mathbf{q}\|_2^2 + \gamma' \sum_{i=1}^{h} \|\mathbf{q}_i\|_2^2 + \gamma_{\text{coord}}\mathcal{L}_{\text{coord}},$$

with coordination regularization weight $\gamma_{\text{coord}} > 0$.

These regularization and optimization techniques are crucial for maintaining stability during training, especially given the exponential nature of graded scaling. Without such controls, gradients involving λ^{q_k} can easily become unstable, either vanishing or exploding, particularly in deep or multi-headed architectures.

7.3. **Convergence and Gradient Stability.** Convergence and gradient stability are critical for ensuring the Exponentially Graded Transformer's training process is robust, particularly given the exponential sensitivity introduced by learnable grading transformations.

Theorem 7.3 (Convergence)**.** *Fix a grading vector* $\mathbf{q}$*. If the loss function* ℓ *is Lipschitz continuous, then gradient descent with sufficiently small step size converges to a stationary point of the Exponentially Graded Transformer's loss.*

Proof. The Exponentially Graded Transformer $\mathcal{EGT}_{\mathbf{q},\lambda}$ is Lipschitz continuous by Prop. 6.13, with constant

$$L \leq L_{\mathcal{T}}(\lambda^{\mathbf{q}_{\max}})^{Lh+3},$$

where $q_{\max} = \max_{k=1,\ldots,d} q_k$. The exponentially graded loss $\mathcal{L}$ is Lipschitz continuous with constant $L_\ell \cdot \max_k \lambda^{q_k}$, as ℓ is Lipschitz with constant L_ℓ. The composite function $\mathcal{L} \circ \bar{\Psi}_{\mathbf{q},\lambda}$ has a Lipschitz gradient with constant proportional to $L \cdot \max_k \lambda^{q_k}$. By classical gradient descent results [14], a step size $\eta < 2/(L \cdot \max_k \lambda^{q_k})$ ensures convergence to a stationary point. □

Proposition 7.4 (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* $\lambda^{\mathbf{q}_{\max}} \ln \lambda$*.*

Proof. The form of the derivative is given by

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial q_k} = \sum_{i=1}^{m}\sum_{j=1}^{d}\left(\lambda^{q_j}\frac{\partial \ell}{\partial \hat{y}_{i,j}} \cdot \frac{\partial \hat{y}_{i,j}}{\partial q_k} + \ell(\hat{y}_{i,j}, y_{i,j}) \ln \lambda \cdot \lambda^{q_k}\delta_{j,k}\right) + 2\gamma q_k. \tag{92}$$

To verify Lipschitz continuity, compute the second derivative:

$$\frac{\partial^2 \mathcal{L}_{\text{total}}}{\partial q_k^2} = \sum_{i=1}^{m} \Big(\lambda^{q_k} (\ln \lambda)^2 \ell(\hat{y}_{i,k}, y_{i,k}) \tag{93}$$

$$+2\lambda^{q_k} \ln \lambda \cdot \frac{\partial \ell}{\partial \hat{y}_{i,k}} \cdot \frac{\partial \hat{y}_{i,k}}{\partial q_k} + \lambda^{q_k} \frac{\partial^2 \ell}{\partial \hat{y}_{i,k}^2} \left(\frac{\partial \hat{y}_{i,k}}{\partial q_k} \right)^2$$

$$+\lambda^{q_k} \frac{\partial \ell}{\partial \hat{y}_{i,k}} \cdot \frac{\partial^2 \hat{y}_{i,k}}{\partial q_k^2} \Big) + 2\gamma. \tag{94}$$

Assuming ℓ is Lipschitz with constant L_ℓ, $\left| \frac{\partial \ell}{\partial \hat{y}_{i,k}} \right| \leq L_\ell$, and $\hat{y}_{i,k}$ depends on q_k through the output layer (Sec. 6.2), which is Lipschitz with constant $\lambda^{\mathbf{q}_{\max}} L_{\text{out}}$. The dominant term is $\lambda^{q_k} (\ln \lambda)^2 \ell(\hat{y}_{i,k}, y_{i,k})$, bounded by $m\lambda^{\mathbf{q}_{\max}} (\ln \lambda)^2 \cdot \text{const}$, yielding a Lipschitz constant proportional to $\lambda^{\mathbf{q}_{\max}} \ln \lambda$. □

Proposition 7.5 (Training Stability for Learned Grades)**.** *Assuming $\|\mathbf{q}_i\|_2, \|\mathbf{k}_j\|_2 \leq C$, the training process with learned $\mathbf{q}$ is stable if the learning rate for q_k satisfies $\eta_q < 1/(\lambda^{\mathbf{q}_{\max}} \ln \lambda)$.*

Proof. From Prop. 7.4, the gradient $\frac{\partial \mathcal{L}_{\text{total}}}{\partial q_k}$ is Lipschitz with constant proportional to $\lambda^{\mathbf{q}_{\max}} \ln \lambda$. For gradient descent updates

$$q_k \leftarrow q_k - \eta_q \frac{\partial \mathcal{L}_{\text{total}}}{\partial q_k}, \tag{95}$$

stability requires $\eta_q < 1/L_g$, where $L_g \propto \lambda^{\mathbf{q}_{\max}} \ln \lambda$. Thus, choosing $\eta_q < 1/(\lambda^{\mathbf{q}_{\max}} \ln \lambda)$ ensures stable updates, preventing divergence due to exponential sensitivity in λ^{q_k}. □

Proposition 7.6 (Sensitivity to λ)**.** *The Exponentially Graded Transformer's training is sensitive to the choice of λ, with the loss gradient's magnitude scaling as $O(\lambda^{\mathbf{q}_{\max}} \ln \lambda)$. A bounded $\lambda \leq \lambda_{\max}$ ensures stable optimization.*

Proof. The gradient $\frac{\partial \mathcal{L}_{\text{total}}}{\partial q_k}$ includes terms like

$$\lambda^{q_k} \ln \lambda \cdot \ell(\hat{y}_{i,k}, y_{i,k}), \tag{96}$$

scaling with $\lambda^{\mathbf{q}_{\max}} \ln \lambda$. From Prop. 6.13, the Exponentially Graded Transformer's Lipschitz constant scales as $(\lambda^{\mathbf{q}_{\max}})^{Lh+3}$. Large λ amplifies gradients, risking instability. Bounding

$$\lambda \leq \lambda_{\max} \tag{97}$$

(e.g., $\lambda_{\max} = 2$) limits the gradient magnitude, ensuring stable optimization with

$$\eta_q < 1/(\lambda_{\max}^{\mathbf{q}_{\max}} \ln \lambda_{\max}).$$

□

7.4. **EGT-Specific Training Challenges.** The Exponentially Graded Transformer introduces novel training challenges due to its exponential scaling and multi-head grading structure. In contrast to earlier models [1, 2], where grading tuples were fixed, here both the global grade vector $\mathbf{q} = (q_0, \ldots, q_{d-1})$ and the per-head vectors $\mathbf{q}_i = (q_{i,0}, \ldots, q_{i,d_k-1})$ are treated as learnable parameters. This improves adaptability but increases the risk of instability in optimization. The following algorithm summarizes a robust training procedure tailored to EGT.

Algorithm 7.1 (Exponentially Graded Training)**.**

***Input**: Training data, initial model parameters θ, grade tuples $\mathbf{q}$, $\mathbf{q}_i$, initial $\lambda = 1$, target $\lambda_{\max}$, total steps T, gradient threshold τ, regularization weights $\gamma, \gamma', \gamma_{coord}$.*

***Note**: All grades $\mathbf{q}, \mathbf{q}_i$ are learnable.*

***Initialize**:*

Set $\mathbf{q} = (q_0, \ldots, q_{d-1})$ using domain-informed values (e.g., $q_k = c \cdot k$, $c > 0$),

Set $\mathbf{q}_i = (q_{i,0}, \ldots, q_{i,d_k-1})$ similarly for each attention head $i = 1, \ldots, h$.

for** each training step $t = 1, \ldots, T$ **do

Update $\lambda_t = 1 + (\lambda_{\max} - 1) \cdot \frac{t}{T}$.

Compute loss components:

$$\mathcal{L}_{coord} = \sum_{i=1}^{h} \left\| \mathbf{q}_i - \frac{1}{h} \sum_{j=1}^{h} \mathbf{q}_j \right\|_2^2,$$

$$\mathcal{L}_{total} = \mathcal{L} + \gamma \|\mathbf{q}\|_2^2 + \gamma' \sum_{i=1}^{h} \|\mathbf{q}_i\|_2^2 + \gamma_{coord} \mathcal{L}_{coord}.$$

Compute gradients $\nabla \mathcal{L}_{total}$ *and apply clipping:*

$$\mathbf{g}' = \begin{cases} \mathbf{g}, & \text{if } \|\mathbf{g}\|_2 \leq \tau, \\ \frac{\tau}{\|\mathbf{g}\|_2} \cdot \mathbf{g}, & \text{otherwise.} \end{cases}$$

Estimate

$$\mathbf{q}_{\max} = \max\left(\max_k q_k, \max_{i,k} q_{i,k} \right)$$

Use learning rate $\eta_q < 1/(\lambda_t^{\mathbf{q}_{\max}} \ln \lambda_t)$ *for grade updates; update* $\theta, \mathbf{q}, \mathbf{q}_i$ *using Adam.*
end for
Output: *Optimized parameters* $\theta, \mathbf{q}, \mathbf{q}_i$.

The components of Algorithm 7.1 are designed to address the following key training challenges:

i) **Grade Initialization**: Domain-informed initialization (e.g., $q_k = c \cdot k$) improves convergence and prevents poor early gradients due to randomly scaled inputs.
ii) **Tuning** λ: Large values of λ amplify important features (Prop. 6.13) but risk instability via terms like λ^{q_k}. Annealing λ_t from 1 to $\lambda_{\max}$ mitigates this risk while preserving expressivity.
iii) **Multi-Head Coordination**: Independent grade vectors $\mathbf{q}_i$ enhance expressivity (Prop. 6.23) but can diverge. The regularization term $\mathcal{L}_{\text{coord}}$ penalizes deviations from the headwise mean, promoting coherence among heads.
iv) **Gradient Control**: Exponentially graded mappings produce gradients sensitive to q_k, requiring gradient clipping to prevent numerical overflow. The learning rate for grades is adaptively bounded by $\lambda_t^{\mathbf{q}_{\max}} \ln \lambda_t$ to ensure stability.

Together, these procedures ensure stable and effective training of EGT models. Empirical validation on tasks such as symbolic regression and structured prediction (cf. Sec. 8) will further illustrate the benefits of adaptive grading over fixed architectures [1, 2].

8. Potential Applications

The graded transformer embeds hierarchical priors via LGT and EGT, enabling principled learning across structured domains. This section outlines potential impact in algebraic geometry, physics, natural language processing (NLP), biological sequence analysis, cross-domain settings, and emerging areas such as graph learning and financial modeling. Each domain exhibits natural grading—via degree, scale, structure, or functional relevance—which the graded transformer exploits to enhance attention efficiency (Prop. 6.10), reduce sample complexity (Prop. 5.16), and improve interpretability. Linear grading suits moderate hierarchies, while exponential grading amplifies deep structures. Mathematical examples illustrate how grading is applied, and new applications highlight the graded transformer's versatility, while acknowledging challenges in grade design.

8.1. **Algebraic Geometry.** Graded structures are central to algebraic geometry, from polynomial rings to moduli spaces and weighted projective varieties. The graded transformer aligns with these by emphasizing basis elements according to algebraic degree or weighted monomial structure, with learnable grades adapting to complex geometries.

i) Modeling moduli spaces and isogenies, where graded invariants govern geometry [9, 15, 16]; exponential grading amplifies high-degree terms.
ii) Computing zeta functions and point counting on hypersurfaces, scaling monomials via grading to reduce complexity (Prop. 5.16).

Mapping discrete degrees to continuous grades remains nontrivial; this is mitigated by learning $\mathbf{q}$ (Sec. 7.4).

8.2. **Physics.** Many physical systems are multiscale, from microscopic energy levels to macroscopic dynamics. Grading emphasizes dominant features consistent with physical intuition (quantum mechanics, fluid dynamics, cosmology).

i) Spectral prediction in quantum systems, scaling by energy eigenvalues to emphasize high-energy orbitals [2]; linear grading for moderate scales.
ii) Turbulence and phase transitions, emphasizing low-wavenumber structures via inverse-frequency grading for stability (Prop. 6.13).

Continuous spectra can leverage learned grades; see Sec. 7.4.

8.3. **Natural Language Processing.** Language exhibits hierarchical structure; words and syntactic roles have varying semantic weight. The graded transformer adjusts attention via learnable grades, encoding structural and sequential hierarchies (Prop. 6.20).

i) Syntactic parsing and semantic role labeling, scaling heads via grading for efficiency (Prop. 6.10).
ii) Dialogue understanding and question answering, elevating intent-bearing tokens with reduced VC dimension (Prop. 5.16).

Adapting grades to syntactic conventions benefits from structured corpora and learned $\mathbf{q}_i$.

8.4. **Biological Sequence Analysis.** Biological sequences contain functionally significant elements embedded in noise. Grading scales these elements, improving sample efficiency (Prop. 5.16) in genomics and proteomics.

i) Gene prediction and variant effect modeling, emphasizing coding regions via exponential grading for deep hierarchies.
ii) Protein modeling and metagenomics, weighting active sites with robustness guarantees (Prop. 6.13).

8.5. **Cross-Domain Applications.** Grading abstracts importance across domains, enabling transfer learning on structured data.

i) Pretraining on symbolic tasks (e.g., Gröbner-basis learning) followed by fine-tuning on NLP or genomics.
ii) Encoding data fabrics as 4D graded structures for flow-aware learning.

8.6. **Robotics.** Robotics systems must balance safety and efficiency in dynamic environments. Graded architectures provide a principled mechanism for prioritizing safety-critical features over throughput or energy goals. For example, collision-avoidance signals from proximity sensors receive higher grades than energy-optimization objectives, ensuring obstacle detection dominates decisions.

In human–robot interaction, high-grade features tied to human presence or gestures override lower-grade efficiency goals, aligning behavior with safety and ethical imperatives. Recent advances, such as logically constrained transformers for safe robotic planning [17], demonstrate how grading can integrate with temporal logic to enforce safety constraints in real-time planning. Furthermore, diffusion-based models like the Robotics Diffusion Transformer [18] could benefit from exponential grading to amplify hierarchical action sequences in foundation models for robotics.

8.7. **Automotive.** In ADAS and autonomous vehicles, certain decisions must carry mathematically provable priority. For example, "do not hit a pedestrian" should outrank "maintain optimal speed." Graded transformers encode such hierarchies by amplifying safety-critical features in attention and decision layers, making them dominant in the output.

This graded prioritization supports interpretable and verifiable AI, offering algebraic guarantees that safety constraints are not overridden by secondary objectives (efficiency, comfort), and thereby aiding regulatory certification. Transformer-based object detection models for self-driving cars [19] can be enhanced with linear grading to prioritize dynamic obstacles, while generative frameworks for pre-crash trajectories [20] leverage exponential grading for multi-scale temporal dependencies in complex driving scenarios.

8.8. **Integration with Symbolic Rules.** A key advantage of graded transformers is their neuro-symbolic nature, allowing seamless integration of continuous, sensor-driven learning with discrete symbolic rules. In automotive contexts, symbolic constraints (traffic laws, right-of-way, ethical directives) can be assigned explicit grades to ensure compliance despite perceptual uncertainty.

In robotics, symbolic goals such as "do not harm humans" are embedded as high-grade features, guaranteeing dominance in learning and inference. This hybrid framework bridges numerical optimization with logical reasoning, yielding a transparent and mathematically rigorous approach to safety-critical decision making. Neuro-symbolic frameworks, such as those integrating reasoning with deep learning for

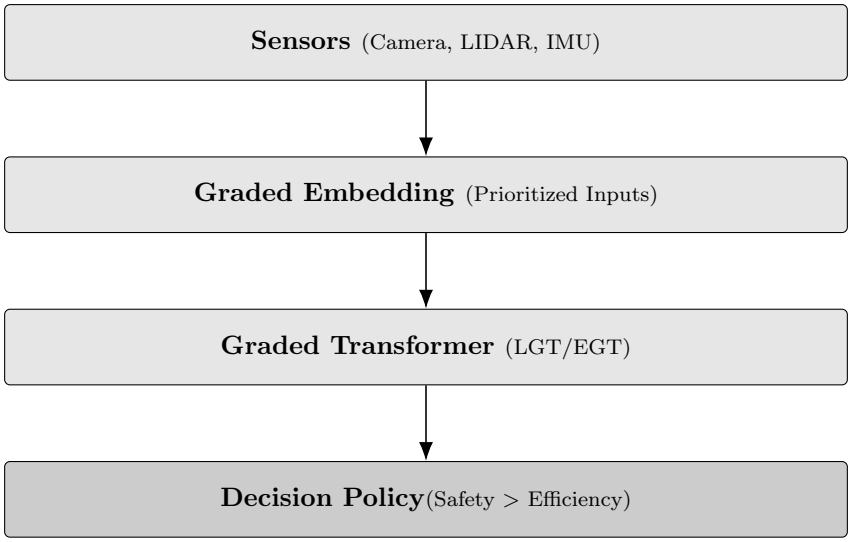


FIGURE 1. Graded transformer pipeline for safety-critical systems. High-grade safety features dominate lower-priority objectives throughout the stack.

enhanced decision-making in autonomous robotics [21], highlight how graded priors can enforce safety in agile control systems [22], while applications in traffic domains emphasize predictive tasks like safety and perception [23].

8.9. **Additional Applications.** The graded approach extends to further hierarchical domains.

i) Graph neural networks, grading nodes by centrality for community detection or molecular modeling.
ii) Financial time-series forecasting, grading significant market events for volatility prediction.

These examples underscore versatility: learned grades adapt to task hierarchies and enhance efficiency and interpretability. Open challenges remain in grade design and transferability; broad empirical validation across these tasks is an important direction for future work.

Remark 8.1. *Certain aspects of the graded transformer framework are the subject of recent patent applications by the author, highlighting its potential for practical deployment in domains such as NLP, robotics, automotive industry, and financial cryptography.*

Remark 8.2. *Several components of the graded transformer framework are the subject of recent patent applications, underscoring its potential for real-world deployment in NLP, robotics, automotive, and financial cryptography.*

REFERENCES

[1] Tony Shaska, *Artificial Neural Networks on Graded Vector Spaces*, Contemporary Math., vol. 781, Amer. Math. Soc., [Providence], RI, American Math. Soc., 2026.
[2] ———, *Graded Neural Networks*, International Journal of Data Science in the Mathematical Sciences (2025), available at 2502.17751. submitted.
[3] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar, *Are transformers universal approximators of sequence-to-sequence functions?*, arXiv preprint arXiv:1912.10077 (2019).
[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, *Attention is all you need*, arXiv preprint arXiv:1706.03762 (2017).
[5] Tony Shaska, *Finsler Metric Clustering in Weighted Projective Spaces*, 2025. submitted.
[6] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun, *Transformers in time series: A survey*, Proceedings of the thirty-second international joint conference on artificial intelligence, 2023, pp. 6778–6786.
[7] Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov, *Transformer dissection: A unified understanding of transformer's attention via the lens of kernel*, Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp), 2019, pp. 4344–4353.
[8] Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet, *A mathematical perspective on transformers*, arXiv preprint arXiv:2312.10794 (2023December).
[9] Elira Shaska, Jorge Mello, Sajad Salami, and Tony Shaska, *Rational points and zeta functions of Humbert surfaces with square discriminant*, NATO Science for Peace and Security Series - D: Information and Communication Security, vol. 66, IOS Press, 2025.

[10] Vladimir N. Vapnik, *Statistical learning theory*, Wiley-Interscience, 1998.
[11] Bahaa E. A. Saleh and Malvin Carl Teich, *Fundamentals of photonics*, 2nd ed., Wiley, 2007.
[12] Diederik P. Kingma and Jimmy Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980 (2014December).
[13] Hyunjik Kim, George Papamakarios, and Andriy Mnih, *The lipschitz constant of self-attention*, Proceedings of the 38th international conference on machine learning, ICML 2021, 18-24 july 2021, virtual event, 2021, pp. 5562–5571.
[14] Yurii Nesterov, *Lectures on convex optimization*, 2nd ed., Springer, 2018.
[15] Adrian Clingher, Andreas Malmendier, and Tony Shaska, *Kummer Surfaces, Isogenies and Theta Functions*, NATO Science for Peace and Security Series - D: Information and Communication Security, vol. 66, IOS Press, 2025.
[16] 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
[17] Parv Kapoor, *Logically constrained robotics transformers for enhanced perception-action planning*, 2024.
[18] Songming Liu et al., *Rdt-1b: a diffusion foundation model for bimanual manipulation*, 2024.
[19] Istiaq Ahmed Fahad, *Automatic vehicle detection using detr: A transformer-based approach for navigating treacherous roads*, 2025.
[20] Xichang Liu, Helai Huang, Jiang Bian, Rui Zhou, Zhiyuan Wei, and Hanchu Zhou, *Generating intersection pre-crash trajectories for autonomous driving safety evaluation using transformer time-series generative adversarial networks*, Engineering Applications of Artificial Intelligence (2025).
[21] Chen Wang, *Imperative learning: A self-supervised neuro-symbolic learning framework for robot autonomy*, The International Journal of Robotics Research (IJRR) (2025).
[22] Sunny Katyara, *Toward general physical intelligence for resilient agile manufacturing automation*, 2025.
[23] Kamal Acharya, *Integrating neurosymbolic ai in advanced air mobility: A comprehensive survey*, 2025.

Department of Mathematics and Statistics,, Oakland University, Rochester, MI 48309
Email address: shaska@oakland.edu