Review and Critical Analysis

Attention Is A Smoothed Cubic Spline

Lai, Z., Lim, L.-H., & Liu, Y. (arXiv:2408.09624, 2024)
https://arxiv.org/abs/2408.09624

1 Summary

This paper provides a rigorous approximation-theoretic characterization of the Transformer architecture. Lai et al. argue that the supremacy of deep learning over classical approximation methods stems from a fundamental shift in function modeling: from summation (linear combinations of basis functions) to composition (nested functions).

The authors establish a mathematical isomorphism between Transformers and spline theory. Specifically, they prove that a ReLU-based Self-Attention mechanism is mathematically equivalent to a multivariate cubic spline (degree 3), while the Feed-Forward Network (FFN) corresponds to a linear spline (degree 1). By invoking the Pierce-Birkhoff conjecture from real algebraic geometry, the paper proposes a "Bidirectional Equivalence": not only are Transformers splines, but every multivariate spline can be represented as a ReLU-based Transformer encoder. This work effectively demystifies the "black box" of attention, framing it as a natural evolution of classical interpolation methods optimized for high-dimensional composition.

2 Methodological Framework

2.1 The Scaling Argument: Composition vs. Summation

The paper creates a dichotomy between classical approximation theory and modern deep learning:

- **Summation (Classical):** Functions are approximated as $F(x) = \sum_{i=1}^{k} c_i \phi_i(x)$. To cover an n-dimensional space with a basis size d per dimension, this requires a tensor product structure, leading to a parameter count of $\Theta(nd^n)$. This exponential growth represents the classical "Curse of Dimensionality."
- **Composition (Deep Learning):** Transformers model functions as $F = F_t \circ \cdots \circ F_1$. The authors demonstrate that for a depth t and embedding dimension n, the parameter count scales as $\Theta(tn^2)$.

The paper posits that Transformers overcome the curse of dimensionality precisely because they construct high-degree splines through composition rather than by exhaustively enumerating a tensor-product grid.

2.2 Degree Arithmetic of Transformer Components

The paper's structural analysis rests on treating the Transformer as a composite function over vector spaces. To quantify the expressivity of these functions, the authors derive the polynomial

degree using two fundamental lemmas regarding the algebra of splines. Let $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^p \to \mathbb{R}^n$ be spline functions of degree d_f and d_g respectively.

1. **Functional Composition (Multiplicative Growth):** The degree of the composition $f \circ g$ is the product of the degrees:
$$\deg(f \circ g) = \deg(f) \times \deg(g) \tag{1}$$
This lemma explains why depth is crucial: stacking layers composes functions, causing the degree to grow exponentially.

2. **Algebraic Interaction (Additive Growth):** If two splines are combined via scalar or matrix multiplication (e.g., an inner product or element-wise weighting), their degrees add:
$$\deg(f \cdot g) = \deg(f) + \deg(g) \tag{2}$$
This lemma governs the internal mechanics of a single Attention head, where Query, Key, and Value vectors interact multiplicatively.

2.2.1 Attention as a Multivariate Cubic Spline (Degree 3)

The Self-Attention mechanism is mathematically deconstructed as a sequence of three operations, transforming the input X into a piecewise cubic manifold. The attention operation is defined as:

$$\text{Attention}(X) = V(X) \cdot g\left(K(X)^\top Q(X)\right) \tag{3}$$

The degree accumulation proceeds in three stages:

1. **Affine Projection (Degree 1):** The learnable projections $Q(X) = W_Q X + b_Q$, $K(X) = W_K X + b_K$, and $V(X) = W_V X + b_V$ are affine transformations. In spline theory, these are globally linear splines of degree 1.

2. **Bilinear Interaction (Degree 2):** The unnormalized similarity score $S(X) = K(X)^\top Q(X)$ represents the pairwise interaction between tokens. Since this involves the matrix product of two degree-1 functions, by the additive rule, the resulting surface is a quadratic spline (degree $1 + 1 = 2$). Geometrically, this introduces curvature into the representation.

3. **Gating and Weighting (Degree 3):** The gating function g (typically ReLU in this theoretical framework) creates the "knots" of the spline but preserves the polynomial degree of the regions it partitions (degree 0 operation on the polynomial order). Finally, the context vector is formed by weighting the values $V(X)$ by the scores. This multiplication of the quadratic scores (degree 2) with the linear values (degree 1) yields a cubic spline (degree $2 + 1 = 3$).

While standard Transformers use Softmax, the authors argue that Softmax is simply a C^∞ (smooth) approximation of the piecewise cubic structure defined by the hard ReLU attention, sharing the same fundamental shape properties.

2.2.2 Encoder-Decoder as a Quintic Spline (Degree 5)

A naive analysis might suggest that composing an encoder and a decoder would multiply their degrees ($3 \times 3 = 9$). However, the authors rigorously prove that a single Encoder-Decoder block implements a quintic spline (degree 5). This reduction is due to the specific topology of Cross-Attention.

Let $\beta(Y)$ be the output of the decoder's masked self-attention (a cubic spline in target Y) and let X be the encoder output (linear in terms of the cross-attention block inputs). The cross-attention mechanism computes:

$$\gamma(X, Y) = V(X) \cdot \text{ReLU}\left(K(X)^\top Q(\beta(Y))\right) \tag{4}$$

Tracing the degree contributions reveals:

- The Query Q transforms the cubic decoder state $\beta(Y)$, preserving degree 3.
- The Key K and Value V transform the encoder state X, which are degree 1 relative to the attention block's operation.
- The interaction $K(X)^\top Q(\beta(Y))$ multiplies a degree-1 term with a degree-3 term, resulting in degree $1 + 3 = 4$.
- The final weighting by $V(X)$ adds one more degree: $1 + 4 = 5$.

This result highlights that the encoder-decoder bridge is an additive coupling of polynomial structures rather than a deep functional nesting, limiting the polynomial complexity to degree 5 per block.

2.3 Exponential Expressivity via Recursive Depth

While individual blocks are low-degree splines (cubic or quintic), the power of the Transformer arises from the recursion of depth. When stacking t layers, the output of layer i becomes the input to layer $i + 1$. This is a functional composition, governed by the multiplicative rule:

$$\deg(\text{Stack}_t) = \deg(\text{Block}) \times \deg(\text{Stack}_{t-1}) \tag{5}$$

For a standard encoder stack where each block is cubic (degree 3), the recurrence is $d_t = 3 \cdot d_{t-1}$ with base case $d_1 = 3$. The solution is:

$$\deg(\text{Transformer}_t) = 3^t \tag{6}$$

This derivation provides a rigorous quantification of the "depth vs. width" debate. Widening the model (increasing dimension n) only increases the number of spline knots (linear capacity). However, deepening the model (increasing t) increases the polynomial degree of the spline exponentially. This allows deep Transformers to approximate highly oscillatory or complex decision boundaries that are mathematically impossible for shallow networks or additive spline models to capture efficiently.

3 Universality: The Inverse Direction

While Sections 3–6 establish that every Transformer implements a spline, Section 7 attempts the rigorous converse: proving that *any* multivariate spline can be implemented by a Transformer. This result establishes a bidirectional equivalence between the architecture and the function class. The proof relies on bridging algebraic geometry and neural network theory via two advanced constructs:

1. **The Veronese Map and Logarithmic Depth:** To represent high-degree polynomials, the authors utilize the Veronese map $\nu_k : \mathbb{R}^n \to \mathbb{R}^{\binom{n+k}{k}}$, which "lifts" input variables into a feature space containing all monomials up to degree k. A key insight of the paper is that this high-dimensional map does not require a massive single layer. Instead, the authors demonstrate

that the degree-k Veronese map can be constructed recursively by composing the quadratic Veronese map ν_2:

$$\nu_{2^t} = \underbrace{\nu_2 \circ \nu_2 \circ \cdots \circ \nu_2}_{t \text{ times}} \tag{7}$$

Since the attention mechanism can compute quadratic interactions (degree 2), a Transformer can implement ν_{2^t} using a stack of t layers. This proves that a Transformer can generate the full polynomial basis of degree k with a depth of only $O(\log k)$. Consequently, any polynomial $p(x)$ becomes a linear function in the lifted space defined by $\nu_k(x)$, allowing the Transformer to "linearize" complex polynomial problems efficiently.

2. **Max-Definability via the Pierce-Birkhoff Conjecture:** The second step requires representing the *piecewise* nature of splines. The authors invoke the Pierce-Birkhoff Conjecture (1956), a major open problem in real algebraic geometry. The conjecture asserts that every continuous piecewise polynomial function (spline) $f : \mathbb{R}^n \to \mathbb{R}$ is "max-definable," meaning it can be expressed as a finite lattice combination of polynomials:

$$f(x) = \max_i \min_j p_{ij}(x) \tag{8}$$

where p_{ij} are polynomials.

This connects directly to the architecture because ReLU networks are naturally universal approximators for max-definable functions (as $\max(a, b) = \text{ReLU}(a - b) + b$). The paper synthesizes these findings in **Corollary 3.9**, proposing a four-way equivalence (conditional on the conjecture):

$$\text{Splines} \iff \text{Transformers} \iff \text{Max-Definable Fns} \iff \text{Linear Splines} \circ \nu_k$$

This implies that the Transformer encoder is not merely a subset of spline functions, but is structurally universal for the entire class of splines, provided the algebraic geometry conjecture holds.

4 Critical Analysis

4.1 Strengths

- **Formalization of "Depth vs. Width":** Deep learning practitioners have long intuited that "depth yields expressivity." This paper quantifies that intuition precisely: depth yields exponential growth in the polynomial degree of the underlying spline function (3^t), whereas width only contributes to the granularity of the partitioning (knots).
- **Demystifying Attention:** By decomposing Attention into affine projections and bilinear products, the paper strips away the "magic" of the architecture, revealing it as a standard tool from approximation theory (the cubic spline) applied in a novel compositional manner.
- **Theoretical Rigor:** The distinction between the degree arithmetic of *composition* versus *interaction* is a subtle but critical insight that explains why encoder-decoder blocks are degree 5 rather than degree 9, preventing an overestimation of the model's complexity.

4.2 Limitations and Theoretical Gaps

- **The "Smoothed" Approximation:** The title refers to "Smoothed" cubic splines, referencing the Softmax function. However, the rigorous degree arithmetic relies heavily on the ReLU

formulation (piecewise polynomials). Softmax is C^∞ (analytic) and strictly speaking has no finite polynomial degree. The equivalence relies on viewing Softmax as a probabilistic approximation of the hard ReLU spline. While intuitively sound, this slightly weakens the algebraic exactness of the proofs when applied to standard Softmax Transformers.

- **Reliance on Conjectures:** The proof of the converse direction (Splines $\implies$ Transformers) rests on the Pierce-Birkhoff conjecture. While this connects the work to deep questions in real algebraic geometry, it renders the "bidirectional equivalence" theoretically conditional. If the conjecture is false, there exist splines that Transformers cannot efficiently represent.
- **Knot Placement and Optimization:** The paper focuses on *representational capacity* (what the function *is*) rather than *optimization dynamics* (how to find it). In spline theory, the placement of knots is crucial. The paper implies knots are determined by the bias terms in the network, but does not address the difficulty of learning optimal knot configurations via gradient descent, which is a non-trivial problem in approximation theory.

5 Conclusion

Lai, Lim, and Liu provide a compelling bridge between the disparate worlds of splines and deep learning. By formalizing the Transformer as a hierarchical composition of cubic splines, they offer a mathematically grounded explanation for the architecture's efficiency in high-dimensional spaces. This work suggests that the Transformer is not merely a successful heuristic, but a natural realization of compositional approximation theory, potentially paving the way for spline-based improvements to modern architectures.