ATTENTION FREE ALTERNATIVE TO LANGUAGE MODELLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Most of the recent Transformer models have attention mechanism as a key component. Attention mechanism is the de-facto standard of transformer model as it enhances performance significantly. Nevertheless, it is well-established that attention mechanisms entail significant computational expenses. This places a restriction on model and input size when computing is limited. Additional limitations of attention mechanisms include the attention sink phenomenon. The primary goal of language model is to have efficient sentence representation. In attention mechanism this goal is achieved through similarity computation of tokens based on a starting representation, which is subsequently refined across different levels. We employ an algebraic approach. Tokens are mapped to a lower dimension first. Token pairs are subsequently generated based on the specified attention window size. The two-dimensional space spanned by the token pairs are encoded through a similarity based encoding approach. Our proposed method, the Algebraic language model (ALM), was evaluated using Wikitext2 data. Our experimental results using a laptop demonstrate improvement of the perplexity score by more than 25% in ALM relative to conventional Transformer models. Beside, ALM has substantially less computational complexity compared to Transformer models.

1 INTRODUCTION

Contemporary language models predominantly employ self-attention as their fundamental primitive. The motivation is grounded in the premise that robust performance requires the model to capture the interdependencies between tokens within a sentence. Self-attention is one of many viable mechanisms for achieving this objective. Given a sequence of hidden states $H \in R^{L\times d}$ (here L is the sequence length and d is dimension), self-attention constructs queries, keys, and values from the input sequence: $Q = HW_Q,\ K = HW_K,\ V = HW_V$ and subsequently computes a $L \times L$ matrix of pairwise similarities through QK^T followed by by a softmax normalization. The resulting weights are then applied to combine the values.

The fundamental objective of effective language model representation is to establish a mapping from tokens such that sentences are transformed to their correct semantic interpretations. In this mapping, tokens that are semantically related should have higher similarity compared to unrelated tokens. Self-attention exemplifies tensor lifting where a hidden vector is projected into a higher-dimensional space. This projection is driven by pairwise interactions. Gradient-decent is employed to learn this lifting operation using training data. Transformer models (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020; Dosovitskiy, 2020) typically incorporate multiple attention heads per layer. The overall network comprising numerous stacked layers. Such a configuration results in substantially larger model sizes. Although the resulting models demonstrate considerable expressiveness, the reliance on self-attention mechanisms renders them largely opaque and consequently less interpretable. Furthermore, self-attention continues to represent a significant computational bottleneck and introduces several undesirable phenomena, including attention sink effects (Gu et al., 2025).

This paper introduces an attention-free model designed to provide enhanced interpretability. Pairwise interactions between tokens are characterized as points within a two-dimensional space on the Grassmann manifold $G(2, r)$. These pairs and derived higher-order constructs are represented as points in $\mathbb{R}^{\binom{r}{2}}$. While Plücker Embedding is conventionally employed in this context, we have

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

adopted a similarity-based embedding approach. Although Plücker Embedding effectively captures relationships among geometric objects such as lines and planes, language models demonstrate superior performance with token similarity mappings compared to alternative geometric mappings. Empirical results validate the efficacy of our methodology. The proposed Algebraic Language Model (ALM) architecture comprises the following components:

1. Tokens are transformed from the embedding space $\mathbb{R}^d$ to a reduced-dimensional space of dimension $\mathbb{R}^r$ via the mapping $W_r : \mathbb{R}^d \to \mathbb{R}^r$, where $r \ll d$. The mapping W_r is learned from training data.
2. Token pairs ($z_t,\ z_{t+\Delta} \in \mathbb{R}^r$ at positions t and $t + \Delta$) within a context window span a 2-dimensional space, which corresponds to a point on the Grassmann manifold $Gr(2, r)$.
3. These subspaces are projected into a finite-dimensional projective space of dimension $\mathbb{R}^{\binom{r}{2}}$ through a dot-product-based mapping.
4. Geometric features are integrated into the original space $\mathbb{R}^d$ via Gated mixing.

We assessed the ALM architecture on the Wikitext-2 dataset (Merity et al., 2016). The ALM model with 7.2M parameters demonstrates a 28% improvement in validation perplexities relative to a comparably-sized Transformer baseline. The ALM mixing mechanism exhibits linear complexity with respect to sequence length, whereas self-attention exhibits quadratic complexity.

The contributions of this paper are:

1. An attention-free architecture designated ALM is introduced. This architecture substitutes self-attention mechanisms with a Causal mixing layer. The mixing layer represents token pair affinity as a point-pair within a 2-dimensional subspace of $Gr(2, r)$ where $r \ll d$. All point pairs are subsequently encoded as points in $\mathbb{R}^{\binom{r}{2}}$. Finally, features are projected back to the original space $\mathbb{R}^d$.
2. Empirical validation on Wikitext-2 demonstrates that an ALM model comprising 7.2M parameters achieves competitive performance relative to a Transformer baseline on Wikitext-2.
3. The asymptotic computational complexity of the Algebraic mixing block exhibits linear scaling with respect to sequence length, thereby surpassing the quadratic complexity characteristic of self-attention mechanisms.

The following sections are organized as following: In Section 2, self-attention is described and contrasted with an alternative based on Grassmann manifold. Working of ALM is explained in Section 3. Experimental setup is described in Section 4. Results are presented in Section 5. Section 5. Section 6 is an exposition of rationale for the proposed approach. Relevant related work has been discussed in Section 7. Section 8 reiterates the contribution of the paper and delineates possible future work.

2 Self-attention and Grassmannian mixing

This section examines the implementation specifics of self-attention and Grassmann mixing, providing a comparative analysis of both approaches.

2.1 Multi-head Self-attention

Multi-head self-attention accepts a sequence of hidden states $H \in \mathbb{R}^{L\times d}$ as its input. Each head h executes the following computation for the mixing operation: $Q_h = HW_Q^h\ K_h = HW_K^h,\ V_h = HW_V^h$, where $W_Q^h,\ W_K^h,\ W_V^h \in \mathbb{R}^{d\times d_h}$, and $d_h = d/H$. Attention weights ($\mathbb{R}^{L\times d_h}$) are derived as A_h through the formula $A_h = softmax\left(\frac{Q_h K^T}{\sqrt{d_h}}\right) V_h$.

The multi-head attention mechanism produces its final output by concatenating the individual head outputs O_h, followed by application of a linear projection to the $\mathbb{R}^d$ dimensional space.

2.2 GRASSMANNIAN MIXING

The Grassmann manifold, denoted as $Gr(k, r)$ or $G_k(\mathbb{R}^r)$, represents the collection of all k-dimensional linear subspaces within $\mathbb{R}^r$. This structure constitutes a smooth and compact manifold characterized by dimension $k(r-k)$ and functions as a projective algebraic variety. Given our objective to represent pairwise (excluding ternary or higher-order) interactions between tokens, we restrict our focus to $k = 2$, whereby $Gr(2, r)$ denotes the manifold of 2-dimensional subspaces possessing dimension $2(r-2)$ in $\mathbb{R}^r$. Grassmannians preserve a streamlined local linear structure of intricate token interactions and efficiently represent token similarity through a fixed-dimensional feature vector subject to established algebraic constraints.

How can we represent a set of k dimensional subspaces (Grassmannian) within a given space? In Plücker embedding, each k-dimensional subspace is mapped to a point in projective space $\mathbb{R}^{\binom{r}{2}}$ according to the following procedure:

1. Given a set of k-dimensional subspaces $U \subset \mathbb{R}^r$ $(u_1, \cdots, u_k)$, we construct the exterior product $u_1 \wedge \cdots \wedge u_k$ in the kth exterior power $\wedge^k \mathbb{R}^r$.

2. In coordinate form, the output is a vector in $\mathbb{R}^{\binom{r}{k}}$ whose components correspond to the $k \times k$ minors of the matrix $[u_1, \cdots, u_k]$.

3. For $k = 2$, when $u, v \in \mathbb{R}^r$ span a 2-dimensional subspace, then $u \wedge v$ is determined by all pairwise determinants $p_{ij} = u_i v_j - u_j v_i,\ 1 < i < j \leq r$.

The image of $Gr(2, r)$ under Plücker embedding does not necessarily encompass the entirety of $\mathbb{R}^{\binom{r}{2}}$ but rather constitutes an algebraic variety within this space, characterized by quadratic Plüucker relations. Output p represents a normalized feature vector.

Semantic features are characterized as a manifold M, and the network functions as a mapping $\Phi : M \to M$. According to classical universal approximation theorems, when a neural network possesses adequate capacity, it can approximate any such mapping. However, conventional networks lack transparency and possess no inherent geometric structure. Through Grassmannian flow, wherein local neighborhoods are encoded into $Gr(2, r)$ followed by Plücker embedding and subsequent transformations, the model progresses through a structured manifold with regulated degrees of freedom; crucially, these properties do not compromise the applicability of the universal approximation theorem.

3 A CAUSAL ATTENTION FREE ALGEBRAIC TRANSFORMER

We shall now provide a comprehensive exposition of the ALM architecture. Our approach adheres to the fundamental structure of a Transformer encoder, yet substitutes each self-attention block with a Causal Algebraic mixing block that accomplishes the following:

1. diminishes hidden states into a reduced-dimensional representation,

2. establishes local pairs and represents them as vectors in $\mathbb{R}^{\binom{r}{2}}$, and

3. reintegrates these geometric characteristics into the original hidden states through gating mechanisms and a feed-forward network.

3.1 TOKEN AND POSITIONAL EMBEDDINGS

Positional embedding has demonstrated considerable efficacy in language modelling applications. Token sequences $(x_1, \cdots, x_L)$ are transformed into $\mathbb{R}^d$ utilizing a learned transformation matrix $W_e \in \mathbb{R}^{V \times d}$ and augmented with positional embeddings $P \in \mathbb{R}^{L_{max} \times d}$: $h_t^0 = W_e(x_t) + P_t,\ t = 1, \cdots, L$, where V denotes the vocabulary size. Our experiments employed $d = 128$ and 256. The resulting sequence $H^0 = (h_1^0, \cdots, h_L^0)$ undergoes subsequent transformation through N Causal Algebraic Transformer layers (we evaluated N=4, 6 and 8 exclusively).

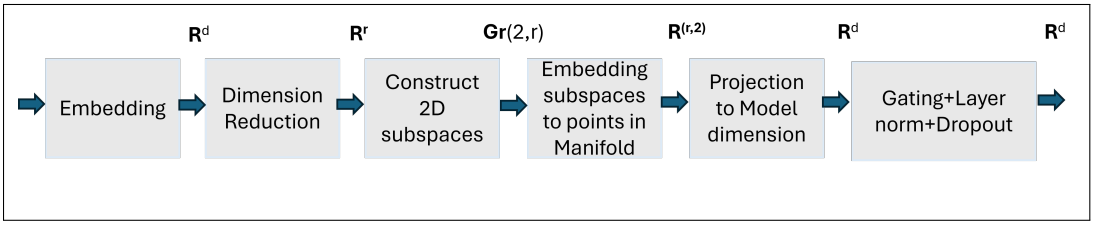

Figure 1: Causal Algebraic Mixing

3.2 Causal Algebraic mixing layer

Each layer receives as input $H \in \mathbb{R}^{L\times d}$ and produces an updated sequence $H \in \mathbb{R}^{L\times d}$. The fundamental operations executed within the layer are:

1. We initially compress each hidden state to a lower-dimensional vector: $z_t = W_{red}h_t + b_{red},\ W_{red} \in \mathbb{R}^{r\times d},\ b_{red} \in \mathbb{R}^r$. Typically $r \ll d$; in our experimental work we employed $r = 16$ and 32. This generates $Z = (z_1, \cdots, z_L) \in \mathbb{R}^{L\times r}$.
2. We establish a collection of window magnitudes (offsets) $W = \{\Delta_1, \cdots, \Delta_m\}$, for instance, $W = \{1, 2, 4, 8, 12, 16\}$.
   With an offset denoted by Δ, token pairs are constructed as $(z_t, z_{t+\Delta})$. Beginning at time t, a total of m pairs are generated in the following manner: $(z_t, z_{t+\Delta_1}), (z_t, z_{t+\Delta_2}), \cdots$. The pairing mechanism exhibits causality, as the value at time t is exclusively paired with values occurring after t.
3. Each pair $(z_t, z_{t+\Delta})$ spans a 2-dimensional subspace within $\mathbb{R}^r$. A vector $p^(\Delta)_t \in \mathbb{R}^{\binom{r}{2}}$ is constructed with entries defined as: $p_{\Delta_{ij}}(t) = z_{t,i}z_{t+\Delta,j} + z_{t,j}z_{t+\Delta,i},\ 1 \le i < j \le r$. It should be noted that this construction *differs from the Plücker embedding*. For numerical stability purposes, p undergoes normalization: $p_t^{(\Delta)} = \frac{p_t^{(\Delta)}}{\max(\|p_t^{(\Delta)}\|^2, \epsilon)}$ for each t and Δ.
4. A linear transformation of features is learned during the training process, mapping from dimension $\binom{r}{2}$ to the model dimension d: $g_t^{(\Delta)} = W_p p_t^{(\Delta)} + b_p,\ W_p \in \mathbb{R}^{\binom{r}{2}\times f}$. The aggregate feature is computed through either simple averaging or weighted averaging: $g_t = \frac{\sum_{\Delta\in W_t} g_t^{(\Delta)}}{|W_t|}$, or $g_t = \frac{\sum_{\Delta\in W_t} \lambda_\Delta g_t^{(\Delta)}}{\sum_{\Delta\in W_t} \lambda_\Delta}$, where $W_t = \{\Delta \in W : t + \Delta \le L\}$ represents the collection of permissible offsets at position t. *With suitable choice of weights, performance of weighted variant is better than the un-weighted version.*
5. The original hidden state and the Grassmann feature are concatenated, and a gating mechanism is computed: $u_t = [ht, gt] \in \mathbb{R}^{2d}$,
   $\sigma_t = \sigma(W_{gate}u_t + b_{gate}), W_{gate} \in \mathbb{R}^{d\times 2d}$. The combined representation is expressed as $h_t^{mix} = \alpha_t \bigodot h_t + (1-\alpha_t) \bigodot g_t$, followed by layer normalization and dropout operations: $h_t = Dropout(LayerNorm(h^{mix}))$.
6. Subsequently, a position-wise feed-forward network is applied, followed by a nonlinear transformation σ (GELU): $\phi_t = W_2\sigma(W_1h_t + b_1) + b_2$, where $W_1 \in \mathbb{R}^{d_{ff}\times d}, W_2 \in \mathbb{R}^{d\times d_{ff}},\ d_{ff} = 4d$. Ultimately, a residual connection and layer normalization are subsequently applied. $h_t^{'} = LayerNorm(h_t + \phi_t)$.

Figure 1 illustrates the architectural design of ALM.

Let us examine the computational time complexity of ALM in comparison to that of a conventional Transformer. For a sequence of length L and model dimension d, a standard multi-head self-attention layer exhibits time complexity $O(Ld^2 + L^2d_{head})$, where d_{head} denotes the per-head dimension. Given a number of heads equal to $\frac{d}{d_{head}}$, the overall complexity becomes $O(Ld^2+L^2d)$.

For the equivalent sequence length and model dimension, projection to the reduced dimension r requires $O(Ldr)$ operations. For a specified t and Δ, the computation of $p_t^{(\Delta)}$ necessitates $O(r^2)$ operations. Therefore, for $m = |W|$ offsets, the total cost is $O(Lmr^2)$. Projection to model space

Table 1: Comparison of ALM with Transformer on WikiText-2

Model	Parameters(M)	Validation Perplexity (% improvement over Transformer)
ALM	7.2	23
Grassmanian	7.2	-27
Transformer	7.26	-
ALM	7.4	28
Grassmanian	7.4	-28
Transformer	7.26	-

($W_p p_t^{(\Delta)}$) requires $O(dr^2)$ operations per pair; consequently, the total cost of projection amounts to $O(Lmdr^2)$. Gating and feed-forward operations require $O(Ld^2)$ operations. When $r \ll d$, the total cost is $O(Ld^2)$ for fixed r and m. Consequently, ALM demonstrates complexity that is linear in L, as opposed to $O(L^2)$ complexity in Transformers.

4 Experimental Setup

We assess the proposed Causal Grassmann architecture using two established NLP benchmarks: Wikitext-2 for language modeling (Merity et al., 2016).

WikiText-2 represents a widely-recognized benchmark dataset employed for training and evaluating language models within natural language processing. The dataset comprises a collection exceeding 100 million tokens sourced from verified Good and Featured articles on Wikipedia. Sequences are constructed from contiguous text segments of predetermined length L. Our primary experiments employ $L = 128$ and $L = 256$. The vocabulary size utilized was $V \approx 30,522$.

For the Transformer model, a standard decoder-only Transformer architecture was implemented, featuring N layers, model dimension d, feed-forward dimension $d_{ff} = 1024$, and multi-head self-attention comprising 4 heads. For ALM, identical parameters N, d, and d_{ff} were employed. The self-attention block in ALM was replaced with a Causal Algebraic mixing block. Based on the number of layers N, models are designated as shallow ($N = 4$) or deep ($N = 8$). The shallow model contains approximately $4.2M$ parameters, whereas deeper models contain $17.7M$ parameters. Experiments were conducted with reduced dimensions $r = 16$ and 32 respectively. The multi-scale windows employed were $W = \{1, 2, 4, 8, 12, 16\}$. Attention score of a token with a token at a distance is less likely than with a token in close neighbourhood. By using weight inversely proportional to the distance, this effect can be achieved. Hence we selected $\lambda_\Delta = \frac{\frac{1}{\Delta}}{\sum \frac{1}{\Delta}}$

All experiments were executed on an HP Omen laptop with the following specifications

Processor Intel Core with 2.90 GHz clock,

RAM 32 GB,

GPU NVIDIA GeForce RTX 5070 GPU (8 GB),

Operating system Windows 11 64 bit.

Both models were trained utilizing the same optimizer name and learning rate schedule. All models underwent training for 10 epochs, with the optimal validation perplexity reported across the training process. The batch size was standardized at 32 for all experiments.

5 Results

Table 1 presents a comprehensive summary of the experimental results. We report parameter counts and optimal validation perplexity achieved over 10 epochs (convergence was attained within this period). For 4-layer models employing a block size of 128 and multi-scale windows W = 1, 2, 4, 8, 12, 16, the ALM/Grassmannian architecture achieves sizes of 7.4M and 7.2M for reduced dimensions of 32 and 16 respectively. The Transformer model with identical 4-layer configuration

and block size of 128 demonstrates a comparable size of approximately 7.26M. The Grassmannian 7.4M model exhibits a perplexity deterioration of 28% relative to the Transformer baseline. The ALM model same size improves perplexity by 28% over Transformer model and 56% over the Grassmannian model. The Grassmannian 7.2M model has higher perplexity 127% than Transformer, while the ALM model reports a lower perplexity than Transformer (77% of Transformer's perplexity - an improvement of 23% over Transformer and 46% over Grassmannian).

The ALM architecture demonstrates perplexity improvements exceeding 25% even for smaller model variants. For larger models, the performance gains become substantially more pronounced.

6 Discussion

The interpretability challenges inherent in Transformer models can be attributed to the attention mechanism. While attention patterns among tokens in the initial layer remain comprehensible and semantically meaningful, the interpretation of this operation becomes increasingly opaque as information propagates through subsequent layers in higher-dimensional spaces. For analytical purposes, assume that information at deeper layers represents phrasal concepts. In such a scenario, should attention operate among phrases rather than between phrases and individual words? What constitutes the optimal interaction window? Consequently, the global (final) representation of an input may not optimally encapsulate the input information.

Conversely, when information flows across a constrained manifold with restricted degrees of freedom, the representation at each layer inherently encapsulates information from the immediately preceding layer. Therefore, the final representation preserves substantial information from prior layers, particularly when constraints are imposed at each processing stage. For instance, representing information in locally linear form rather than through complex paradigms facilitates tracing information flow. Consequently, we deliberately constrain the relevant degrees of freedom within a finite-dimensional, mathematically rigorous manifold:

1. The reduced states $z_t \in \mathbb{R}^r$ represent local orientations within a lower-dimensional representation space.
2. Pairs $(z_t, z_{t+\Delta})$ establish points on the Grassmannian $\mathrm{Gr}(r, 2)$; these points are characterized by Plücker vectors possessing a fixed dimensionality of $\binom{r}{2}$.
3. The mixing mechanism is restricted to local perturbations of these low-rank subspaces, rather than unrestricted modifications of an $L \times L$ tensor.
4. The embeddings demonstrate consistency across different layers.
5. The transformation can be examined and studied utilizing mathematical frameworks derived from differential geometry and algebraic geometry. Outcomes from the model exhibit enhanced interpretability.

From the standpoint of approximation theory, one may conceptualize a sequence model as approximating an operator Φ defined on a semantic manifold $M \subset \mathbb{R}^d$. Universal approximation theorems guarantee that, subject to reasonable assumptions, neural networks possess the capacity to approximate such operators with arbitrary precision. Nevertheless, the approximation theorem remains indifferent to the underlying geometric structure. Through the incorporation of Grassmann manifolds, we introduce supplementary geometry-informed inductive bias without compromising approximability.

7 Related Work

Transformers have established themselves as a robust alternative, demonstrating proficiency in managing both local and long-range dependencies while facilitating parallelized training (Tay et al., 2022). Notable models include GPT-3 (Brown et al., 2020) and ChatGPT (Kocon et al., 2019). During inference, Transformers exhibit time and space complexity of $O(L^2 d)$ and $O(L^2 + Ld)$ respectively, where L denotes sequence length and d represents the feature dimension. Reformer (Kitaev et al., 2020) introduces two methodologies to enhance Transformer efficiency. It substitutes dot-product attention with locality-sensitive hashing-based attention, reducing complexity from

$O(L^2)$ to $O(L \log L)$. Additionally, it employs reversible residual layers in place of standard residuals, enabling single-instance activation storage during training rather than N instances, where N signifies the number of layers. Consequently, space complexity is diminished to $O(L \log L + Ld)$.

Performer (Choromanski et al., 2020) presents a linear-complexity alternative to conventional softmax attention in Transformer architectures. It approximates standard softmax attention kernels through random feature maps, thereby reducing time and memory requirements from quadratic to linear: $O(Ld^2 \log d)$ and $O(Ld \log d + d^2 \log d)$ respectively.

cosFormer (Qin et al., 2022) represents an efficient linear transformer architecture. To attain linear complexity while preserving essential softmax properties, ReLU is utilized for attention matrix non-negativity, and a non-linear re-weighting mechanism is implemented via a cosine distance function. This approach reduces the standard transformer's $O(L^2)$ time and space complexity to $O(Ld^2)$ and $O(Ld + d^2)$ respectively.

AFT (Zhai et al., 2021) formulates attention alternatively as $Attn^+(W, K, V)_t = \frac{\sum_{i=1}^{t} e^{w_{t,i}+k_i} \bigodot v_i}{\sum_{i=1}^{t} e^{w_{t,i}+k_i}}$ where $w_{t,i} \in \mathbb{R}^{L \times L}$ represents learned pair-wise position biases. AFT-full exhibits $O(L^2 d)$ and $O(Ld)$ complexity respectively.

Receptance Weighted Key Value (RWKV) model (Peng et al., 2023) integrates the capabilities of RNNs and Transformers while mitigating their respective limitations. RWKV addresses the memory constraints and quadratic computational scaling inherent in Transformers (Katharopoulos et al., 2020) through efficient linear scaling, while preserving the Transformer's advantageous characteristics, including parallelized training capabilities and robust scalability.

RWKV reformulates the attention mechanism by employing a linear attention variant that substitutes conventional dot-product token interactions with more efficient channel-directed attention. This approach, implemented without approximation, achieves minimal computational and memory complexity; $O(Ld)$ and $O(d)$ respectively.

LLaMA (Touvron et al., 2023) and Chinchilla (Hoffmann et al., 2022) demonstrate the efficacy of Transformers in natural language processing. Nevertheless, the self-attention mechanism's quadratic complexity presents significant computational and memory challenges for applications requiring extended sequences and operating under resource constraints.

Substantial scholarly investigation (Zhao et al., 2026) seeks to address the quadratic computational burden inherent in self-attention mechanisms, incorporating approaches such as linearized or kernelized attention, sparse and localized attention configurations, and memory-augmented or retrieval-based systems. These techniques generally:

- reduce or sparsify the QK^T operation,
- restrict attention to local regions or predetermined patterns, or
- delegate portions of contextual information to external storage or buffers.

Despite these innovations, they preserve the core mechanism: the model still computes (or approximates) an $L \times L$ matrix representing pairwise interaction weights. Our approach represents a fundamental departure: we eliminate attention mechanisms entirely and examine whether a geometric mixing operation derived from Grassmann flows (Chong, 2025) can fulfill this role.

State-space models and analogous frameworks represent sequences as signals governed by linear dynamical systems, frequently combined with non-linear output mechanisms. Such models exhibit remarkable efficacy in extended-context scenarios with computational complexity linear in sequence length and maintain substantial alignment with control theory and signal processing disciplines.

Grassmann mixing aligns with state-space models in maintaining a structured latent representation that evolves sequentially; nevertheless, the conceptual orientation diverges:

- State-space models emphasize temporal development of latent representations; their geometric characteristics remain largely implicit.
- Grassmann mixing prioritizes geometric transformation within representation space, with temporal advancement realized through causal masking.

These complementary viewpoints offer valuable perspectives, and hybrid architectures incorporating state-space temporal mechanisms alongside Grassmann constraints on latent representations constitute a valuable direction for future research.

Grassmann flows (Chong, 2025) failed to enhance perplexity contrary to the paper's assertions. Our examination demonstrated that the model could be optimized in specific dimensions, which we have integrated into ALM, specifically:

- Plücker embedding proves ineffective for capturing sentence semantic representation. We have implemented a symmetric embedding methodology wherein lines and spaces from $Gr(2, r)$ are projected onto points in $\mathbb{R}^{\binom{r}{2}}$.
- Differentiated treatment of two-dimensional spaces in $Gr(2, r)$ based on temporal distance between token pairs demonstrates superior effectiveness in sentence representation.

Computational approaches on non-Euclidean spaces, including manifolds, have demonstrated considerable efficacy across numerous applications. These methodologies typically embed data onto curved manifolds where distance metrics more faithfully capture underlying structural properties relative to Euclidean geometry. Consequently, manifolds, encompassing Grassmann manifolds, have been employed in diverse machine learning contexts including subspace clustering, low-rank approximation, and metric learning, where they characterize collections of subspaces. However, their utilization as a primary mixing mechanism in sequence models remains substantially underexplored. The outcomes from Grassmann flow (Chong, 2025) were unsatisfactory. Our contribution presents an innovative embedding approach as a substitute for Plücker encoding alongside differential weighting mechanisms for token pairs. Empirical findings demonstrate considerable potential.

Let us examine the soundness of the propose embedding.

- The resulting coordinates exhibit symmetric ($p_{ij} = p_{ji}$), whereas in standard Plücker coordinates are antisymmetric ($p_{ij} = -p_{ji}$). The proposed encoding maps lines in $Gr(2, r)$ to points in $\mathbb{R}^{\binom{r}{2}}$.
- Plücker coordinates characterize a 2-dimensional subspace on the Grassmannian variety. The additive embedding presented in this work, by contrast, characterizes a point on the secant variety of the Veronese variety.
- In the new encoding, the notion of direction or orientation between the two vectors is lost, while the standard Plücker coordinates preserve the orientation. In fact, the notion of orientation here is not the same as the causal relation between token pairs. In the new encoding too computation involving z_t includes $z_{t+\Delta}$ and not $z_{t-\Delta}$.

Our contribution does not represent a state-of-the-art result. Additional investigation is warranted in the following domains:

- First, performance evaluation of models incorporating larger parameter counts may demonstrate the effectiveness of the proposed architecture.
- Second, comparative assessment of the architecture against established large language model benchmarks should be undertaken.
- Third, thorough hyperparameter optimization of the weights λ may yield additional insights into language modeling.

8 Conclusion and Future Work

Self-attention constitutes a fundamental component in Transformer architectures. The enhancement of self-attention computational efficiency has garnered considerable attention throughout recent years. The majority of existing research concentrates on optimizing the computational implementation of attention mechanisms. We reconsider the fundamental purpose underlying attention and propose alternative approaches distinct from conventional attention. Manifolds have demonstrated substantial utility and advantages in machine learning applications. The proposed manifold-based

alternative (ALM) offers not only computational efficiency but is also grounded in rigorous mathematical foundations. This enhancement promotes improved model interpretability and consequently, greater explainability.

The following represent potential directions for subsequent investigation:

1. Investigate higher-dimensional subspaces exceeding two dimensions for token interaction analysis. For k-wise interaction (with $k > 2$) among tokens, the subspace $Gr(k, r)$ can be mapped to points in $\mathbb{R}^{\binom{r}{k}}$.
2. Examine alternatives beyond Plücker coordinates and the method presented in this work, which may incorporate constraints aligned with linguistic semantics.
3. Derive optimal weights for different Δ theoretically or empirically.
4. Investigate hybrid architectures integrating other frameworks such as state space models, which may prove advantageous. The evolution of manifold structure across layers in recurrent neural networks is more likely to align appropriately when suitable constraints are established.
5. Evaluate ALM performance on video content and other multimodal applications.

AI USE STATEMENT

We have used generative AI tools for the purpose of enhancing the article's readability through suggestions. However, we have refrained from utilizing generative AI tools for any other purposes.

ETHICS STATEMENT

There is no ethics violation to the best of authors' knowledge.

REPRODUCIBILITY STATEMENT

A hyperlink directing to an anonymous, downloadable source code repository shall be provided as supplementary materials.

REFERENCES

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Zhang Chong. Attention is not what you need. *arXiv preprint arXiv:2512.19428*, 2025.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations*, volume 2025, pp. 97114–97144, 2025.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Jan Kocon, Piotr Miłkowski, and Monika Zaśko-Zielińska. Multi-level sentiment analysis of polemo 2.0: Extended corpus of multi-domain consumer reviews. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 980–991, 2019.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, et al. Rwkv: Reinventing rnns for the transformer era. In *Findings of the association for computational linguistics: EMNLP 2023*, pp. 14048–14077, 2023.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. *arXiv preprint arXiv:2202.08791*, 2022.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer. *arXiv preprint arXiv:2105.14103*, 2021.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, et al. A survey of large language models. *Frontiers of Computer Science*, 20(12):2012627, 2026.