OmegaMGT: Isometric Dynamics and Asymptotic Dimensional Contraction in Single-Highway Transformers

Anonymous authors
Paper under double-blind review

Abstract

Current Transformer architectures dominate natural language processing under a constant dimensionality paradigm, which imposes a computational cost that grows linearly with network depth ($\mathcal{O}(L)$). This dimensional redundancy demands massive hardware capabilities for parameter storage and context maintenance, severely limiting their operational scalability and prohibiting the local deployment of state-of-the-art models in resource-constrained environments. In this work, we introduce the ***Omega Minimal Gated Transformer (OmegaMGT)***, a deep neural architecture that challenges this standard through a geometric dimensionality reduction topology. We propose two fundamental innovations: (1) a *per-instance* spectral token compression mechanism based on Principal Component Analysis (PCA), which aligns the sequence length with its intrinsic informational content; and (2) a *single-highway* structure based on the Minimal Gated Unit (MGU), which replaces additive residual connections with a modulated deep recurrence. Through rigorous theoretical analysis, we demonstrate that OmegaMGT decouples the asymptotic *time* complexity of inference from model depth, achieving an $\mathcal{O}(1)$ bound as $L \to \infty$, while activation memory grows only through the compressed attention term ($\mathcal{O}(L\omega^2)$) rather than the full model width. We further prove that, at initialization and under (near-)isometric projections, the gradient norm is preserved across depth, providing a stable pathway for backpropagation in massively deep regimes. This approach offers a theoretical framework for building massively deep yet computationally bounded language models, pointing toward LLM inference under tight compute budgets.

Keywords: Efficient Transformer, Geometric dimensionality reduction, Minimal Gated Unit (MGU), Spectral compression, Deep Recurrent Neural Networks.

1 Introduction

Large Language Models (LLMs) have redefined the state-of-the-art (SOTA) in artificial intelligence, but their practical adoption faces a critical bottleneck: computational inefficiency. Current architectures, descendants of the original Transformer (Vaswani et al., 2017) and more recently the GPT architecture (Radford et al., 2018), maintain constant dimensionality (d_{model}) across dozens or even hundreds of layers (L) (Minaee et al., 2024). On the other hand, from a linguistic perspective, it has been established that complex propositions can be understood even when words are omitted, as evidenced by phenomena such as ellipsis or Cloze tests (Taylor, 1953). While these principles have inspired NLP techniques like *token pruning*, these approaches face underestimated risks of losing relevant linguistic information (Kim & Cho, 2021; Wen et al., 2025). Both factors invariably lead to linear memory consumption with respect to depth (L) and context window size (N), which precludes the local execution of SOTA models on edge devices (smartphones, standard laptops, IoT) with limited resources or without specialized hardware.

This has led us to hypothesize that maintaining full dimensionality (d_{model} and N) is structurally redundant for two fundamental reasons: (1) As information flows through the network, attention representation abstracts into long-term relationships, reducing the need for bandwidth in deep processing (Vig & Belinkov, 2019). (2) Geometrically, the distribution of embeddings is inherently anisotropic (Ethayarajh, 2019); since

information variance concentrates in privileged directions, the token matrix possesses a low intrinsic dimensionality, allowing for aggressive dimensional reductions in the feature space without compromising information integrity. Although dimensionality reduction techniques like Principal Component Analysis (PCA) theoretically demonstrate the viability of compressing these latent representations, the structural challenge to materialize this reduction has been significant: residual connections (skip connections), vital for gradient flow and training stability, require identical dimensions between input and output (He et al., 2016).

Effectively addressing these structural limitations, we present OmegaMGT, a theoretical architecture designed to maximize computational efficiency. Our proposal integrates two complementary strategies: first, a *dimensionality reduction based on explained variance* through PCA (Hotelling, 1933), designed to compress the initial redundancy of the embeddings; and second, a *progressive geometric reduction*, where the width of the layer decays by a factor r. Crucially, to sustain the flow of information across these variable dimensions—where standard residual connections are inapplicable—we implement a **Block State (BS)** mechanism based on the Minimal Gated Unit (MGU) (Zhou et al., 2016) within a *single-highway* structure.

The main contributions of this article are as follows:

- **Geometric architecture:** We propose a new architecture in which spatial complexity converges to $\mathcal{O}(d_{model}^2)$ independently of depth, as the total parameter count follows a geometric series of convergence.
- **Spectral compression via PCA:** We integrate a dimensionality reduction strategy based on explained variance that exploits the anisotropic nature of embeddings, projecting input data into its intrinsic dimensionality to eliminate redundancy before deep processing.
- **MGU-based state propagation:** We adapt the MGU cell (Zhou et al., 2016) to act as a vertical transport mechanism for gradients between Transformer blocks of different sizes, effectively replacing additive skip connections.
- **Theoretical stability guarantee:** We provide an analytical derivation showing that, despite aggressive dimensionality reduction, gradient magnitude is preserved ($\delta BS_{l-n} \approx \delta BS_l$), mitigating the vanishing gradient problem in deep networks.

Article Organization. The remainder of the article is organized as follows: Section 2 reviews related work on efficient transformers and activation mechanisms. Section 3 formalizes the OmegaMGT architecture, detailing the use of PCA for spectral compression of the token space, geometric reduction, and the Block State mechanism. Section 4 presents the theoretical analysis of gradient stability and computational complexity. Section 5 addresses the technical and theoretical limitations of the present work and analyzes future lines of research. Finally, Section 6 concludes the article.

2 Related Work

2.1 Efficient Transformers

Since the introduction of the original Transformer (Vaswani et al., 2017), the quadratic computational cost of self-attention ($\mathcal{O}(N^2)$) has motivated numerous variants aimed at optimizing resource usage. Approaches such as **Reformer** (Kitaev et al., 2020) utilize *Locality-Sensitive Hashing* (LSH) to reduce complexity, while **Linformer** (Wang et al., 2020) employs low-rank projections to approximate the attention matrix. Although these architectures succeed in mitigating complexity with respect to sequence length (N), they typically maintain a constant model dimension (d_{model}) across all layers. Our approach, OmegaMGT, differs fundamentally by addressing inefficiency in the spatial dimension (d_l), systematically reducing embedding redundancy layer by layer rather than approximating the attention mechanism itself.

2.2 Recurrent Neural Networks and Gating Mechanisms

Gating mechanisms are fundamental for controlling information flow in recurrent neural networks (RNNs), such as LSTMs and GRUs. Specifically, the **Minimal Gated Unit (MGU)** (Zhou et al., 2016) was

proposed as an efficient simplification of the GRU (Chung et al., 2014), coupling the input and forget gates into a single mechanism. In this work, we repurpose the MGU architecture: instead of using it for temporal recurrence (along the sequence), we adapt it for *depth recurrence* (along the layers L). This allows the MGU to manage the vertical gradient flow and act as a dynamic bridge between layers of different dimensions, compensating for the elimination of static residual connections. Closest to our depth-wise gating is the Highway Network (Srivastava et al., 2015), which introduced learnable transform and carry gates to enable the training of very deep feed-forward stacks. OmegaMGT differs in two respects: (i) it contracts the layer width geometrically rather than preserving a constant dimensionality across depth, and (ii) it replaces the additive carry gate with an MGU-style single gate whose internal projection order is reordered to bridge otherwise incompatible dimensions (see Equation 9).

2.3 State Space Models and Hybrid Architectures

Recently, architectures such as **RWKV** (Peng et al., 2023) and **Mamba** (Gu & Dao, 2023) have challenged the dominance of pure Transformers. These models, frequently based on state space models (SSMs) or linear attention, combine parallelizable training with linear complexity inference. While they achieve efficiency regarding sequence length, they generally rely on maintaining large and constant hidden states to preserve context. OmegaMGT proposes an orthogonal and complementary optimization direction: while models like Mamba optimize N, OmegaMGT implements a geometric collapse of the spatial dimension d_{model}. This structural reduction offers memory savings that are independent of the utilized attention mechanism.

2.4 Dimensionality Reduction and Spectral Analysis

Several studies have demonstrated that language model *embeddings* are anisotropic and occupy a narrow cone in the vector space, suggesting they can be significantly compressed (Ethayarajh, 2019; Mu et al., 2018). Post-processing techniques based on PCA have been used to reduce the size of static embeddings (Raunak et al., 2019). OmegaMGT integrates this principle directly into the architecture: we define an initial projection (Omega Matrix) based on the cumulative explained variance to determine the optimal dimensionality with respect to the input length, eliminating spectral token redundancy before the information enters the deep layers.

3 Methodology

We define OmegaMGT as a deep *single-highway* neural architecture, composed of a stack of L blocks indexed by $l \in \{1, \ldots, L\}$. Preserving the idea of standard Transformers that maintain a single flow of information, OmegaMGT unifies the attention flow based on recurrent neural network architecture into a single latent state variable BS that evolves in depth.

Let $BS_0 = \Omega^\top \in \mathbb{R}^{\omega \times d_1}$ be the initial projected input (derived from spectral compression). The model imposes a structural constraint of *geometric dimensional reduction*, forcing the latent space to contract progressively.

3.1 Geometric Dimensionality Reduction

The backbone of OmegaMGT is its convergent topology. We define a reduction factor $r > 1$. The dimensionality d_l of the l-th layer follows the progression:

$$d_l = \left\lfloor \frac{d_{model}}{r^{l-1}} \right\rfloor \tag{1}$$

This contraction implies that the projection matrices within each block must adapt dynamically. The dimension transition is initially executed in the *Feed-Forward Network* (FFN) corresponding to the attention of each block, which acts as a bottleneck. Let $W_1^{(l)}$ and $W_2^{(l)}$ be the weights of the attention FFN in layer l; their dimensions are:

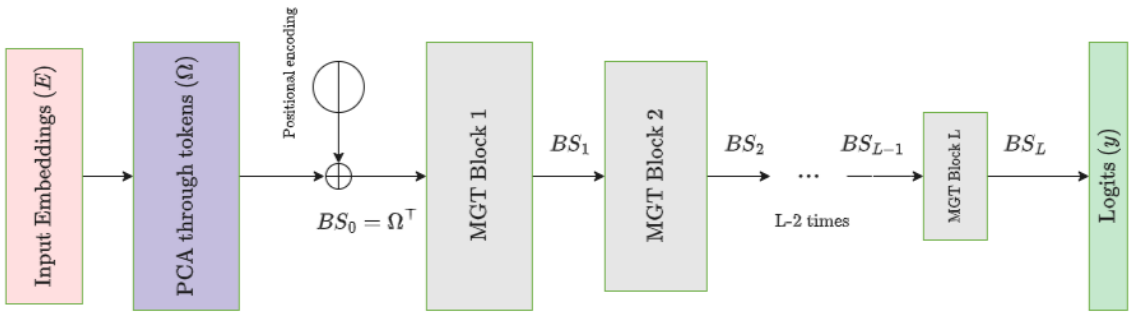

Figure 1: General representation of the OmegaMGT architecture demonstrating the geometric dimensionality reduction d_l for each MGT cell.

$$W_1^{(l)} \in \mathbb{R}^{d_l \times 4d_l}, \quad W_2^{(l)} \in \mathbb{R}^{4d_l \times d_{l+1}} \tag{2}$$

Thus, the FFN receives a representation in the d_l space and generates an *update proposal* in the reduced d_{l+1} space.

3.2 Spectral Token Compression via PCA

Prior to the deep architecture, we apply a *per-instance* preprocessing to align the sequence with its intrinsic dimensionality, exploiting the anisotropy of the embeddings (Ethayarajh, 2019).

Let $E \in \mathbb{R}^{N \times d_{model}}$ be the embedding matrix. We center the data (E_c) and compute the covariance matrix over the token space:

$$\Sigma = \frac{1}{d_{model}} E_c E_c^\top \in \mathbb{R}^{N \times N} \tag{3}$$

We perform an eigenvalue decomposition on Σ (Hotelling, 1933), selecting the ω components that explain the cumulative variance a, a threshold hyperparameter (e.g., $a = 0.95$). We project the embeddings onto this orthogonal basis $Z \in \mathbb{R}^{N \times \omega}$:

$$\Omega = E_c^\top Z \in \mathbb{R}^{d_{model} \times \omega} \tag{4}$$

The initial state of the model is defined as the transpose of this projection:

$$BS_0 = \Omega^\top \in \mathbb{R}^{\omega \times d_{model}} \tag{5}$$

3.3 The OmegaMGT Block: Recurrent Highway

The central innovation of this approach is the **pure depth recurrence**, translating the mechanisms of recurrent networks into a sequential scheme along depth. Unlike RNN recurrence, the weights are *not* shared across layers (each block l owns its own parameters $W^{(l)}$); the recurrence is structural—carried by the state BS—rather than parametric. In block l, attention is not computed over a temporal variable, but directly over the previous memory state BS_{l-1}.

The operational flow to obtain the new state $BS_l \in \mathbb{R}^{\omega \times d_{l+1}}$ from $BS_{l-1} \in \mathbb{R}^{\omega \times d_l}$ is as follows:

3.3.1 Attention over Memory

The *Multi-Head Attention* (MHA) mechanism (Vaswani et al., 2017) extracts contextual features by querying the previous state. The matrices Q, K, V of layer l are derived from BS_{l-1}:

$$A_l = \text{MHA}(BS_{l-1}) \quad \text{where } Q, K, V \sim BS_{l-1} W^{(l)}_{\{q,k,v\}} \tag{6}$$

3.3.2 Update Proposal (Local Variable)

The attention output is processed through the dimensional reduction FFN (Equation 2). This generates X_l, a transient local variable that exists only within the block to propose new features in the target dimension d_{l+1}:

$$X_l = \phi(A_l W_1^{(l)}) W_2^{(l)} \in \mathbb{R}^{\omega \times d_{l+1}} \tag{7}$$

3.3.3 Integration via MGU (State Update)

Finally, we use the *Minimal Gated Unit* to merge the proposal X_l with the historical memory BS_{l-1}. Since BS_{l-1} and X_l have different dimensions ($\omega \times d_l$ vs $\omega \times d_{l+1}$), the parameter matrices U of the MGU (Zhou et al., 2016) act as alignment projectors.

First, we compute the forget gate F_l:

$$F_l = \sigma(X_l W_F^{(l)} + BS_{l-1} U_F^{(l)}) \tag{8}$$

Then, the candidate state $\widehat{BS}_l$:

$$\widehat{BS}_l = \phi(X_l W_{BS}^{(l)} + F_l \odot (BS_{l-1} U_{BS}^{(l)})) \tag{9}$$

Remark. *Note an intentional deviation in Equation 9 with respect to the standard formulation of the MGU cell (Zhou et al., 2016), where typically the gate modulates the state before the projection (i.e., $U(F \odot H_{prev})$). In OmegaMGT, due to the direct dimensional incompatibility between the memory $BS_{l-1} \in \mathbb{R}^{\omega \times d_l}$ and the gate $F_l \in \mathbb{R}^{\omega \times d_{l+1}}$, the algebraic order is inverted: first, the previous state is projected into the target reduced-dimensional space via $BS_{l-1} U_{BS}^{(l)}$, thus enabling its compatibility for modulation through the Hadamard product with F_l.*

The final block state is updated via interpolation:

$$BS_l = (J - F_l) \odot (BS_{l-1} U_{proj}^{(l)}) + F_l \odot \widehat{BS}_l \tag{10}$$

Dimensions of the MGU matrices: To support geometric reduction, the weight matrices are defined as:

$$W_F^{(l)}, W_{BS}^{(l)} \in \mathbb{R}^{d_{l+1} \times d_{l+1}} \quad \text{(Operate in the target space)}$$
$$U_F^{(l)}, U_{BS}^{(l)}, U_{proj}^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}} \quad \text{(Project from the source space)}$$

3.4 Output Projection

At the end of the last block L, the prediction depends exclusively on the final accumulated state.

$$P(y | BS_0 = \Omega^\top) = \text{softmax}(BS_L W_{out}) \tag{11}$$

Where $W_{out} \in \mathbb{R}^{d_L \times |V|}$.

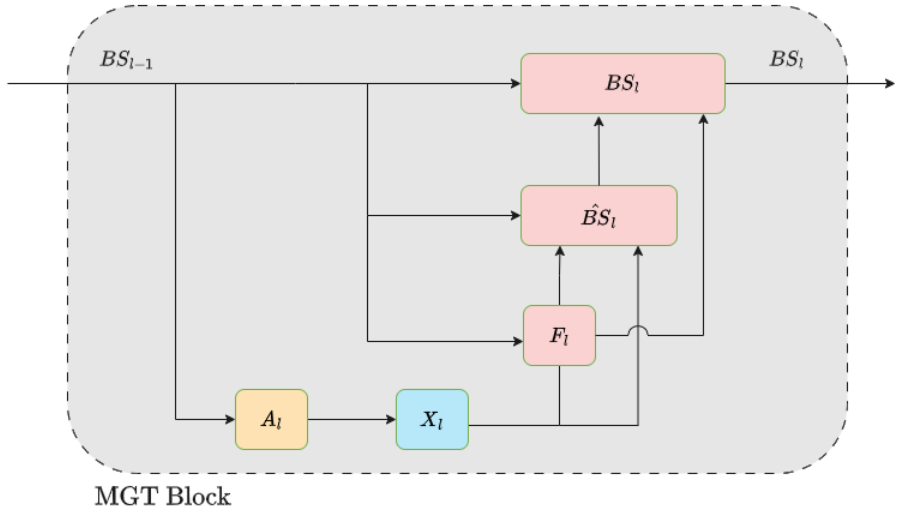

Figure 2: Architecture of a MGT cell

4 Theoretical Analysis

This theoretical analysis is built upon a set of formal definitions in order to validate the mathematical feasibility of the model. We focus on demonstrating two fundamental guarantees that distinguish OmegaMGT: first, that the 'single-highway' mechanism preserves the magnitude of the signal through the depth, mitigating gradient vanishing; and second, that the geometric reduction imposes a convergent upper bound on the computational complexity, offering a significant advantage over the constant-dimension models reviewed in the literature.

4.1 Gradient Analysis

4.1.1 Initial Definitions

Let $\mathcal{L}$ be the scalar loss function computed at the end of the architecture (layer L). To analyze training stability, we evaluate the behavior of the gradient backpropagated from a layer l to a previous layer $l-n$. The goal is to determine if the gradient magnitude is preserved, that is, if the norm $\left\|\frac{\partial \mathcal{L}}{\partial BS_{l-n}}\right\|_2$ remains proportional to $\left\|\frac{\partial \mathcal{L}}{\partial BS_l}\right\|_2$, avoiding the exponential decay characteristic of vanishing gradients.

Making an approximation over the immediately preceding layer $l-1$, we then define the gradient for this layer through the additivity of two expressions:

$$\frac{\partial \mathcal{L}}{\partial BS_{l-1}} = \frac{\partial BS_l}{\partial BS_{l-1}} \frac{\partial \mathcal{L}}{\partial BS_l} + \left(\frac{\partial X_l}{\partial BS_{l-1}}\right)^\top \frac{\partial BS_l}{\partial X_l} \frac{\partial \mathcal{L}}{\partial BS_l} \tag{12}$$

The gradient calculation for the previous block bifurcates into two convergent paths: the *indirect path* that crosses the attention mechanism ($BS_{l-1} \overset{\partial}{\leftarrow} X_l \overset{\partial}{\leftarrow} BS_l$) and the *direct path* of the recurrent state ($BS_{l-1} \overset{\partial}{\leftarrow} BS_l$). Below, we formalize the contribution of the indirect path to subsequently formalize the direct path.

4.1.2 Decomposition of the Attention Path

We define the sequence of intermediate operations within the *Feed-Forward Network* (FFN) as:

$$Z_l := A_l W_1^{(l)}, \quad U_l := \phi(Z_l), \quad X_l = U_l W_2^{(l)}. \tag{13}$$

Remark. *Note the notational distinction between the intermediate activation U_l and the projection matrices of the MGU mechanism ($U_F^{(l)}, U_{BS}^{(l)}$).*

Calculating the total differentials and establishing the linear dependence with respect to dBS_{l-1}, we obtain:

$$dZ_l = (dA_l)\, W_1^{(l)} \tag{14}$$

$$dU_l = \phi'(Z_l) \odot dZ_l \tag{15}$$

$$dX_l = dU_l\, W_2^{(l)} \tag{16}$$

Where $\phi'(Z_l)$ denotes the derivative of the activation function applied element-wise. By composing these terms, the variation of the proposal X_l is expressed as:

$$dX_l = \big(\phi'(Z_l) \odot ((dA_l)\, W_1^{(l)})\big) W_2^{(l)} \tag{17}$$

Since dA_l represents the sensitivity of the attention tensor with respect to the previous state BS_{l-1}, we define the attention Jacobian operator $\mathcal{J}_{\text{att}}^{(l)}$ as the linear transformation:

$$\mathcal{J}_{\text{att}}^{(l)} : dBS_{l-1} \in \mathbb{R}^{\omega \times d_l} \mapsto dA_l \in \mathbb{R}^{\omega \times d_l}$$

Which is explicitly defined as:

$$\mathcal{J}_{\text{att}}^{(l)}[dBS_{l-1}] = \left(\frac{1}{\sqrt{d_k}} \mathcal{J}_{\text{softmax}}^{(l)} \left[(dBS_{l-1} W_q^{(l)}) K^\top + Q (dBS_{l-1} W_k)^\top \right] \right) V + S(dBS_{l-1} W_v^{(l)}) \tag{18}$$

Remark. *The complete and detailed derivation of this operator, as well as of the previously undefined terms present in it, are available in Appendix A.*

Using this definition, we construct the composite operator $\mathcal{M}_l\big[dBS_{l-1}\big]$, which encapsulates the total gradient through the attention branch and FFN:

$$\mathcal{M}_l\big[dBS_{l-1}\big] = \big(\phi'(Z) \odot ((\mathcal{J}_{\text{att}}^{(l)}\big[dBS_{l-1}\big])\, W_1^{(l)})\big) W_2^{(l)} \quad \text{where} \quad \mathcal{M}_l\big[dBS_{l-1}\big] = \frac{\partial X_l}{\partial BS_{l-1}} dBS_{l-1} \tag{19}$$

Remark. *From the perspective of computational functional analysis, the operator $\mathcal{M}_l$ acts as a linear transformation over the matrix space. In this context, the notation $\mathcal{M}_l^\top$ is formally used to denote the adjoint operator with respect to the Frobenius inner product, $\langle A, B \rangle_F = tr(A^\top B)$. By definition, this operator satisfies the duality property $\langle \mathcal{M}_l[A], B \rangle_F = \langle A, \mathcal{M}_l^\top[B] \rangle_F$, justifying its continuous use in the formulation of the gradients.*

Therefore, applying the adjoint rule for reverse matrix differentiation (Giles, 2008), it is demonstrated that:

$$dBS_{l-1} \;=\; \mathcal{M}_l^\top\big[dX_l\big] \tag{20}$$

This is standard backpropagation from the attention mechanism + FFN, with important changes contingent on the elimination of residual connections and the dimensionality reduction of the model. In this formulation,

the magnitude of the gradient propagated by the operator $\mathcal{M}_l\big[dBS_{l-1}\big]$ depends explicitly on the spectral norms of $W_1^{(l)}$ and $W_2^{(l)}$, along with that of $\phi'(Z)$. Formally, the induced norm of the operator can be bounded by the following submultiplicative inequality:

$$\|\mathcal{M}_l\|_2 \leq \|\mathcal{J}_{\text{att}}^{(l)}\|_2 \|W_1^{(l)}\|_2 \|W_2^{(l)}\|_2 \max|\phi'(Z)| \tag{21}$$

These spectral factors are crucial for bounding the Lipschitz constant of the layer and, therefore, guaranteeing the numerical stability of the system by avoiding the explosion or vanishing of the gradient during deep training (Pascanu et al., 2013; Miyato et al., 2018).

Although controlling the Lipschitz constant by restricting spectral norms was originally introduced to stabilize min-max dynamics in the adversarial training of Generative Adversarial Networks (GANs) (Goodfellow et al., 2014; Arjovsky et al., 2017), its underlying mathematical principle is critical and directly extrapolable to the OmegaMGT architecture. In standard Transformer-based LLMs, gradient explosion or vanishing is heuristically mitigated through the use of additive residual connections (He et al., 2016; Vaswani et al., 2017), which act as identity shortcuts. By dispensing with such connections in OmegaMGT in favor of a *single-highway* propagation based on MGU, the gradient flow is entirely contingent on composite matrix transformations across L layers. Consequently, limiting the spectral norm of these operators (as is done in the discriminator of a GAN) becomes the fundamental mechanism to ensure that projections neither exponentially amplify nor attenuate the gradient, thus guaranteeing the theoretical and numerical viability of training in a massively deep regime.

4.1.3 Differential Analysis of the MGU Mechanism (Direct Path)

To complete the analysis of the gradient flow, we decompose the internal dynamics of the *Block State*. Let the auxiliary variables of the forward pass be defined by:

$$\begin{aligned} F_l = \sigma(Z_F) : Z_F &:= X_l W_F^{(l)} + BS_{l-1} U_F^{(l)} \\ \widehat{BS}_l = \phi(Z_{BS}) : Z_{BS} &:= X_l W_{BS}^{(l)} + F_l \odot (BS_{l-1} U_{BS}^{(l)}) \end{aligned} \tag{22}$$

Applying total differentiation and establishing the first-order linear dependence with respect to perturbations in the input and the previous state, we obtain the differentials of the gates:

$$\begin{aligned} dF_l &= \sigma'(Z_F) \odot \big((dX_l) W_F^{(l)} + (dBS_{l-1}) U_F^{(l)}\big) \\ d\widehat{BS}_l &= \phi'(Z_{BS}) \odot \big((dX_l) W_{BS}^{(l)} + dF_l \odot (BS_{l-1} U_{BS}^{(l)}) + F_l \odot ((dBS_{l-1}) U_{BS}^{(l)})\big) \end{aligned} \tag{23}$$

To isolate the contributions of each path, we define two sensitivity operators. First, we introduce the operator $P_l\big[dX_l\big]$, which captures the sensitivity of the next state with respect to the local proposal X_l:

$$P_l\big[dX_l\big] = \frac{\partial BS_l}{\partial X_l} dX_l \tag{24}$$

Given the interdependence established in Equation 19, the projection of this operator onto the attention path is expressed as:

$$P_l\big[\mathcal{M}_l\big[dBS_{l-1}\big]\big] = \frac{\partial BS_l}{\partial X_l} \left(\frac{\partial X_l}{\partial BS_{l-1}}\right)^\top dBS_{l-1} \tag{25}$$

We analyze the system's response by momentarily assuming the previous state to be constant ($dBS_{l-1} = \mathbf{0}$). Under this condition, the partial variations reduce to:

$$
\begin{aligned}
dF_l\Big|_X &= \sigma'(Z_F) \odot \big((dX_l)W_F^{(l)}\big) \\
d\widehat{BS}_l\Big|_X &= \phi'(Z_{BS}) \odot \Big((dX_l)W_{BS}^{(l)} + dF_l\Big|_X \odot (BS_{l-1}U_{BS}^{(l)})\Big) \\
dBS_l\Big|_X &= dF_l\Big|_X \odot (\widehat{BS}_l - BS_{l-1}U_{proj}^{(l)}) + F_l \odot d\widehat{BS}_l\Big|_X
\end{aligned}
\tag{26}
$$

Substituting the partial differentials into the state update equation, we obtain the explicit form of the operator P_l:

$$
\begin{aligned}
P_l\big[\mathcal{M}_l\big[dBS_{l-1}\big]\big] = &\big(\sigma'(Z_F) \odot \big((dX_l)W_F^{(l)}\big)\big) \odot (\widehat{BS}_l - BS_{l-1}U_{proj}^{(l)}) \\
&+F_l \odot \Big(\phi'(Z_{BS}) \odot \big(\big((dX_l)W_{BS}^{(l)} + dF_l\Big|_X \odot (BS_{l-1}U_{BS}^{(l)})\big)\Big)
\end{aligned}
\tag{27}
$$

Similarly, we define the operator $Q_l\big[dBS_{l-1}\big]$ to model the purely recurrent sensitivity (memory):

$$
Q_l\big[dBS_{l-1}\big] = \frac{\partial BS_l}{\partial BS_{l-1}} dBS_{l-1} \tag{28}
$$

Now assuming the local proposal to be constant ($dX_l = \mathbf{0}$), we isolate the history-dependent terms:

$$
\begin{aligned}
dF_l\Big|_{BS} &= \sigma'(Z_F) \odot \big((dBS_{l-1})U_F^{(l)}\big) \\
d\widehat{BS}_l\Big|_{BS} &= \phi'(Z_{BS}) \odot \Big(dF_l\Big|_{BS} \odot (BS_{l-1}U_{BS}^{(l)}) + F_l \odot \big((dBS_{l-1})U_{BS}^{(l)}\big)\Big) \\
dBS_l\Big|_{BS} &= -dF_l\Big|_{BS} \odot (BS_{l-1}U_{proj}^{(l)}) + (J - F_l) \odot \big((dBS_{l-1})U_{proj}^{(l)}\big) \\
&\quad + dF_l\Big|_{BS} \odot \widehat{BS}_l + F_l \odot d\widehat{BS}_l\Big|_{BS}
\end{aligned}
\tag{29}
$$

Unifying the terms from Equation 29 and rearranging algebraically, the recurrent operator takes the form:

$$
Q_l\big[dBS_{l-1}\big] = (J - F_l) \odot \big((dBS_{l-1})U_{proj}^{(l)}\big) + dF_l\Big|_{BS} \odot (\widehat{BS}_l - BS_{l-1}U_{proj}^{(l)}) + F_l \odot d\widehat{BS}_l\Big|_{BS} \tag{30}
$$

Finally, by the principle of linear superposition of total differentials, the complete state variation is expressed as the sum of the sensitivities:

$$
dBS_{l-1} = P_l\big[\mathcal{M}_l\big[dBS_{l-1}\big]\big] + Q_l\big[dBS_{l-1}\big] \tag{31}
$$

This allows us to formulate the master backpropagation equation for OmegaMGT defined in partial differential form in Equation 12. Grouping the adjoint operators and defining the descending gradient $\delta BS_l = \frac{\partial \mathcal{L}}{\partial BS_l}$, the recursive update rule is:

$$
\delta BS_{l-1} = \big[Q_l^\top + \mathcal{M}_l^\top P_l^\top\big]\delta BS_l \tag{32}
$$

4.1.4 Conditions for Asymptotic Stability and Norm Preservation

To guarantee the effective propagation of the gradient through the depth of the architecture, it is necessary to establish two critical structural conditions on the defined operators.

Condition 1: Memory Regime (Gate Saturation) We analyze the behavior of the system at the limit where the network prioritizes memory retention, that is, when the forget gate tends to full aperture

($F_l \to \mathbf{0}$). It is fundamental to specify that this limit does not describe the state of the network after convergence, since a permanent suppression of the update would nullify the model's capacity to learn from context and reduce it to a sequence of iterative linear projections. On the contrary, this analysis is proposed as a theoretical worst-case bound and represents an ideal initialization regime. By mathematically guaranteeing the gradient flow under this extreme condition, we ensure that the architecture possesses a stable pathway for error backpropagation in the early phases of training, before the gates learn to dynamically modulate the integration of new information. Under this premise, the asymptotic properties of the activation functions and their derivatives are:

$$\begin{aligned} \sigma(Z_F) &\to \mathbf{0}, \\ \sigma'(Z_F) &\to \mathbf{0}, \\ dF_l|_X &\to \mathbf{0}, \\ (J - F_l) &\to J. \end{aligned}$$

Consequently, the input sensitivity operator P_l tends to vanish due to its multiplicative dependence on $\sigma'(Z_F)$ and F_l. This implies a natural suppression of the indirect path (Attention/FFN) in favor of the recurrent path. Simplifying Equation 32 under this regime:

$$\delta BS_{l-1} \approx \left[Q_l^\top\right] \delta BS_l \tag{33}$$

Substituting the gate limit into the definition of the recurrent operator (Equation 30), the dominant term reduces to the linear projection:

$$Q_l \left[dBS_{l-1}\right] \approx (dBS_{l-1}) U_{proj}^{(l)} \quad \Longrightarrow \quad Q_l \approx U_{proj}^{(l)} \tag{34}$$

Condition 2: Spectral Isometry The previous simplification leads to the second necessary structural constraint. In order for the gradient magnitude to remain unaltered when transitioning between disparate dimensions ($d_{l+1} \to d_l$), the matrix $U_{proj}^{(l)}$ must act as an *isometric projection*. This requires the transformation to be semi-orthogonal, preserving the Euclidean norm of the projected vectors in the subspace of interest.

Extrapolating this linear behavior backwards along n consecutive blocks, error propagation is modeled as:

$$\delta BS_{l-n} \approx \delta BS_l \left(\prod_{k=0}^{n-1} U_{proj}^{(l-k)} \right) \tag{35}$$

Given that the central objective is to avoid both vanishing and exploding gradients, we analyze the spectral norm of the product chain in Equation 35:

$$\|\delta BS_{l-n}\|_2 \approx \|\delta BS_l\|_2 \left(\prod_{k=0}^{n-1} \left\| U_{proj}^{(l-k)} \right\|_2 \right) \tag{36}$$

If the isometry condition on $U_{proj}^{(l)}$ is satisfied (i.e., its singular values are close to 1), the product of the norms converges to unity:

$$\|\delta BS_{l-n}\|_2 \approx \|\delta BS_l\|_2 \tag{37}$$

This result demonstrates that, under the constraints of isometric initialization and operation in a memory regime, the OmegaMGT architecture preserves the gradient magnitude across L layers, theoretically mitigating the *vanishing gradient* problem despite the geometric dimensionality reduction. We emphasize that

this guarantee is established at initialization and under exact isometry of $U_{proj}^{(l)}$: once the forget gates F_l become active during training, the recurrent operator Q_l departs from $U_{proj}^{(l)}$, and norm preservation in the trained regime is an empirical question that we leave to future validation (Section 5).

4.2 Computational Complexity Analysis

In this section, we present a detailed analysis of the computational cost of the proposed block, denoted as $\mathcal{O}_{total}$. For this purpose, we assume that the cost of activation functions and element-wise operations (Hadamard product) is linear and negligible compared to the dominant complexity of matrix multiplications.

Preliminarily, we define the notation and the critical variables of the system:

1. ω: Length of the token sequence after compression via Principal Component Analysis (PCA).
2. d_l: Dimensionality of the feature space associated with the component BS_{l-1}, corresponding to the input of the MGU block.
3. d_{l+1}: Reduced dimensionality associated with the components BS_l and X_l, where $d_{l+1} < d_l$ according to Equation 1.
4. r: Dimensional reduction factor.

4.2.1 Computational Cost of the Transformer Module

We begin by examining the cost of the linear projections for the matrices Q, K, V. This computation is determined by the matrix product between the state BS_{l-1} and the weights $W_{\{q,k,v\}}^{(l)}$, resulting in the following costs:

$$\text{Time} \rightarrow 3\mathcal{O}(\omega d_l^2) \quad \text{Space} \rightarrow 3\mathcal{O}(d_l^2)$$

Next, the attention mechanism A is computed through the product of the softmax distribution with the value matrix V, which implies:

$$\text{Time} \rightarrow \mathcal{O}(\omega^2 d_l) \quad \text{Space} \rightarrow \mathcal{O}(\omega^2)$$

Subsequently, the data flow enters the *feed-forward* network. Initially, the attention matrix A is operated with $W_1^{(l)}$ and the result is projected via the transformation ϕ:

$$\text{Time} \rightarrow \mathcal{O}(\omega d_l^2) \quad \text{Space} \rightarrow \mathcal{O}(d_l^2)$$

Next, the product of the output of ϕ with the matrix $W_2^{(l)}$ is calculated:

$$\text{Time} \rightarrow \mathcal{O}(\omega d_l d_{l+1}) \quad \text{Space} \rightarrow \mathcal{O}(d_l d_{l+1})$$

Consequently, the aggregate computational cost derived from the Transformer component is:

$$\text{Time} \rightarrow \mathcal{O}(\omega d_l^2 + \omega^2 d_l) \quad \text{Space} \rightarrow \mathcal{O}(d_l^2)$$

4.2.2 Computational Cost of the MGU Unit

For the forget gate F_l, the costs are derived from the addition operations with different complexities, resulting in:

$$\text{Time} \rightarrow \mathcal{O}(\omega d_l^2) + \mathcal{O}(\omega d_l d_{l+1}) \quad \text{Space} \rightarrow \mathcal{O}(d_l^2) + \mathcal{O}(d_l d_{l+1})$$

Similarly, the cost associated with calculating the candidate state $\widehat{BS}_l$ is defined as:

$$\text{Time} \rightarrow \mathcal{O}(\omega d_l^2) + \mathcal{O}(\omega d_l d_{l+1}) \quad \text{Space} \rightarrow \mathcal{O}(d_l^2) + \mathcal{O}(d_l d_{l+1})$$

Finally, for the update of the state BS_l, only the product with the projection matrix $U_{proj}^{(l)}$ is computed, yielding:

$$\text{Time} \rightarrow \mathcal{O}(\omega d_l d_{l+1}) \quad \text{Space} \rightarrow \mathcal{O}(d_l d_{l+1})$$

Therefore, the total complexity attributable to the MGU unit is determined by:

$$\text{Time} \rightarrow \mathcal{O}(\omega d_l d_{l+1}) \quad \text{Space} \rightarrow \mathcal{O}(d_l d_{l+1})$$

4.2.3 Global Complexity Analysis and Convergence

Regarding the spatial complexity associated with the activation memory (RAM), the cost per block is composed of two factors: the temporary storage of the attention matrix ($\mathcal{O}(\omega^2)$). By evaluating this global cost across the L blocks of the MGT architecture, we obtain the summation:

$$Cost_{space_act} = \sum_{l=1}^{L} \left(C\omega^2 \right) \tag{38}$$

Consequently, the total spatial complexity of activations is given by:

$$\mathcal{O}_{total}(L\omega^2)$$

Similarly, the prevalent computational time cost per block is $\mathcal{O}(\omega d_l^2 + \omega^2 d_l)$. Extending the calculation over the L blocks:

$$Cost_{time} = \sum_{l=1}^{L} C(\omega d_l^2 + \omega^2 d_l) \tag{39}$$

Substituting the dimensional reduction terms from Equation 1 and assuming the continuous upper bound by omitting the floor function to facilitate the asymptotic analysis, the expression is transformed into:

$$Cost_{time} = \sum_{l=1}^{L} C\Big[\omega\Big(\frac{d_{model}}{r^{l-1}}\Big)^2 + \omega^2\Big(\frac{d_{model}}{r^{l-1}}\Big)\Big] \tag{40}$$

Rearranging the summation:

$$Cost_{time} = C\Big[\omega d_{model}^2 \sum_{k=0}^{L-1} \Big(\frac{1}{r^2}\Big)^k + \omega^2 d_{model} \sum_{k=0}^{L-1} \Big(\frac{1}{r}\Big)^k\Big] \tag{41}$$

Since $|r| > 1$, both geometric series are convergent. Consequently, the total computational time cost is asymptotically:

$$\mathcal{O}_{total}(\omega d_{model}^2 + \omega^2 d_{model})$$

The complexity analysis demonstrates that, thanks to the progressive dimensionality reduction, the OmegaMGT architecture exhibits a fundamental property: the computational cost of inference in time is **strictly independent of the model's depth (L)** in terms of asymptotic complexity (Big-O). Regarding the activation memory (space), the cost of the latent states also converges to a constant, limiting the linear growth with depth exclusively to the compressed attention matrix ($L\omega^2$). This theoretically makes feasible the construction of significantly deeper models without incurring the resource overflow characteristic of standard Transformers.

Preprocessing cost. The per-instance spectral compression incurs a one-off cost per forward pass that is separate from the deep stack. Forming the token covariance $\Sigma = \frac{1}{d_{model}} E_c E_c^\top \in \mathbb{R}^{N \times N}$ costs $\mathcal{O}(N^2 d_{model})$ in time, and its eigendecomposition costs $\mathcal{O}(N^3)$ in time and $\mathcal{O}(N^2)$ in space. This term is independent of the depth L but scales with the original sequence length N, and must therefore be amortized against the $\mathcal{O}(L)$ savings obtained in the deep stack. For long contexts, this motivates replacing the exact decomposition with truncated or randomized SVD, which we discuss as future work.

Table 1: Comparison of global computational complexity (evaluated across the L layers). N denotes the input sequence length, ω the sequence length post-PCA, d_{model} the initial model dimension, and L the total number of blocks/layers. Spatial complexity evaluates the activation memory requirements (RAM). For OmegaMGT, the tabulated costs correspond to the deep stack; they exclude the one-off per-instance PCA preprocessing, which adds $\mathcal{O}(N^3)$ in time (see the discussion above).

Model	**Global Time**	**Global Space (Activations)**
Transformer	$\mathcal{O}(LN^2 d_{model} + LNd_{model}^2)$	$\mathcal{O}(LN^2 + LNd_{model})$
Reformer	$\mathcal{O}(LN \log N d_{model})$	$\mathcal{O}(LN \log N + LNd_{model})$
Performer	$\mathcal{O}(LNd_{model}^2 \log d_{model})$	$\mathcal{O}(LNd_{model} \log d_{model} + Ld_{model}^2 \log d_{model})$
Linear Transformers	$\mathcal{O}(LNd_{model}^2)$	$\mathcal{O}(LNd_{model} + Ld_{model}^2)$
OmegaMGT (Ours)	$\mathcal{O}(\omega^2 d_{model} + \omega d_{model}^2)$	$\mathcal{O}(L\omega^2)$

5 Discussion and Limitations

The OmegaMGT architecture represents a paradigm shift from a constant-depth design towards a geometrically convergent topology. Through the theoretical analysis developed in the preceding sections, we have demonstrated that it is possible to decouple the model's depth (L) from its asymptotic computational complexity, transforming the linear growth $\mathcal{O}(L)$ characteristic of standard Transformers into a bounded convergent series $\mathcal{O}(1)$. This property suggests that OmegaMGT could scale to profound depths (e.g., several dozens or even hundreds of layers with an r very close to 1) with a fraction of the computational and memory budget required by current models.

Furthermore, the integration of the modified MGU mechanism as a unique gradient pathway offers a structural solution to the vanishing gradient problem, eliminating the need for additive residual connections that force the maintenance of a static dimensionality. The preservation of the gradient norm under partial isometry conditions theoretically validates the feasibility of training this deep network.

5.1 Current Limitations

Despite the theoretical guarantees presented, it is imperative to acknowledge the inherent limitations of the current state of this research:

- **Empirical Validation:** The present work establishes the mathematical and algorithmic foundations of the architecture. Currently, OmegaMGT is a theoretical model without a large-scale reference implementation in frameworks such as PyTorch or JAX. The experimental verification of scaling laws (Kaplan et al., 2020) and convergence on real language modeling tasks (e.g., WikiText-103, C4) remains a critical pending task.
- **Hyperparameter Sensitivity:** The stability of the model depends heavily on the choice of the reduction factor r and the PCA variance threshold a. Inadequate values could collapse the representation capacity of the model before reaching the final layers.
- **Preprocessing overhead:** The $\mathcal{O}(N^3)$ per-instance eigendecomposition required by the PCA compression may dominate for long input sequences, partially offsetting the depth-related savings that motivate edge deployment. Approximate spectral methods (e.g., truncated or randomized SVD) are a necessary practical direction to make the preprocessing scale.

5.2 Future Work

To transition from theory to a robust implementation, we outline four research directions that address the identified stability and initialization challenges:

5.2.1 Selective Component Normalization

The application of normalization techniques such as *Layer Normalization* (Ba et al., 2016) must be strategic. We propose applying normalization exclusively to the internal components of the block (such as the input to the gates F_l), facilitating the initial convergence towards $F_l \to \mathbf{0}$ (memory regime). This decision is grounded in recent findings regarding *pre-RMSNorm* configurations (Zhang & Sennrich, 2019), which have proven crucial for stabilizing the variance of activations in deep architectures that dispense with standard additive residual connections or replace them with dynamic gating mechanisms (Gu & Dao, 2023; Guo et al., 2024). However, directly normalizing the block state (BS_l) in the main propagation path must be avoided, as this could break the linear isometry property necessary for gradient transitivity demonstrated in Equation 36.

5.2.2 Stochastic Gate Initialization

The initialization conditions for the forget gate F_l are non-trivial in a variable dimension scheme. It is necessary to investigate initialization schemes that favor a gradual opening of the gate (e.g., a negative bias in the sigmoid), allowing the gradient to flow freely in the early stages of training before the network learns to selectively forget information.

5.2.3 Isometry Regularization (Soft Orthonormality)

To empirically guarantee the theoretical condition $Q_l \approx U_{proj}^{(l)}$ and norm preservation, we propose integrating an auxiliary regularization term into the global loss function, based on the Parseval penalty (Cisse et al., 2017):

$$\mathcal{L}_{total} = \mathcal{L}_{model} + \lambda \sum_{l=1}^{L} \left\| (U_{proj}^{(l)})^\top U_{proj}^{(l)} - \mathbf{I} \right\|_F^2 \tag{42}$$

Where λ is a weighting coefficient. This term penalizes the deviation of the projection matrix U_{proj} from orthonormality, forcing an isometric projection that stabilizes gradient propagation without the need for hard constraints such as SVD decomposition.

Nevertheless, from an optimization perspective, it is fundamental that the parameter λ is subject to an annealing schedule during training (Bowman et al., 2016). Imposing a strict orthonormality penalty from the initial phases can excessively restrict the hypothesis space, inducing a collapse in the network's capacity

to extract highly complex representations (Bansal et al., 2018). Therefore, we propose dynamically diluting the restriction of λ as the model converges. This strategy allows prioritizing gradient stability in the early stages (*warm-up phase*) and, simultaneously, maximizing the expressivity of the projections in the final stages of learning.

5.2.4 Rotary Positional Embeddings (RoPE) in Compressed Space

A critical consideration arises regarding the nature of the compressed input matrix Ω. Since the PCA-based compression reduces the sequence from N original tokens to ω *eigen-tokens*, the strict absolute positional order is blurred, as each *eigen-token* represents a global linear combination of the original sequence. In this context, the implementation of *Rotary Positional Embeddings* (RoPE) (Su et al., 2024) is proposed not to encode token indices, but to establish a relational structure between these principal components.

By applying RoPE over the compressed sequence dimension ω, the model can effectively capture relative dependencies within the latent space. This approach allows the architecture to recover the underlying structural topology lost during PCA, enabling the self-attention mechanism to weigh the importance of *eigen-token* interactions based on their relative distance in the compressed manifold rather than their feature-wise variance.

6 Conclusion

In this work, we have introduced OmegaMGT, an architectural proposal that breaks with the orthodoxy of isometric latent spaces in large language models. By integrating a geometric dimensionality reduction with a deep recurrent state propagation mechanism, we have formulated a structural solution to the computational inefficiency problem in Transformers.

Our theoretical analyses yield two main results. First, we demonstrate that, at initialization and under near-isometric projections, the gradient norm can be preserved without relying on identity residual connections, using instead a *gradient pathway* based on semi-orthogonal projections and saturated MGU gates. Second, and perhaps more importantly, we prove that the *time* cost of OmegaMGT converges to a finite geometric series, so that inference time becomes asymptotically independent of the network depth for large L, while activation memory grows only through the compressed attention term ($\mathcal{O}(L\omega^2)$) rather than the full model width.

Although large-scale empirical validation remains a line of future work, the mathematical guarantees presented suggest that the redundancy in current architectures is massive and dispensable. OmegaMGT not only proposes a lighter model, but also posits a fundamental hypothesis: that intelligence in language models can emerge from progressive hierarchical compression, bringing natural language processing closer to the physical and energetic constraints of consumer hardware.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. *International conference on machine learning*, pp. 214–223, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL `https://arxiv.org/abs/1810.09102`.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, 2016. URL `https://arxiv.org/abs/1511.06349`.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Moustapha Cisse, Piotr Bojanowski, Grave Edouard, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pp. 854–863. PMLR, 2017.

Kawin Ethayarajh. How contextual are contextualized word representations? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4509–4518, 2019.

Mike B Giles. Collected matrix derivative results for forward and reverse mode algorithmic differentiation. Technical report, Oxford University, Computing Laboratory, 2008.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, volume 27, 2014.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Yuxin Guo, Yihao Yue, Yunhao Ni, Yizhou Ruan, Wenjun Wu, and Lei Huang. Enjoy your layer normalization with the computation efficiency of rmsnorm. *OpenReview preprint*, 2024. URL `https://openreview.net/forum?id=hwiPhiKMGM`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Gyuwan Kim and Kyunghyun Cho. Length-adaptive transformer: Train once with length drop, use anytime with search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6501–6511, August 2021.

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

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Jiaqi Mu, Pramod Viswanath, and Sumit Bhat. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*, 2018.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pp. 1310–1318. PMLR, 2013.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, 2023.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. URL `https://api.semanticscholar.org/CorpusID:49313245`. OpenAI Technical Report.

Vikas Raunak, Vivek Gupta, and Florian Metze. Effective dimensionality reduction for word embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pp. 235–243, 2019.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Wenhan Bo, and Yunfeng Cao. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

W. L. Taylor. Cloze procedure: A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433, 1953.

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

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*, 2019.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? *arXiv preprint arXiv:2502.11501*, 2025.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL `https://arxiv.org/abs/1910.07467`.

Gugbing Zhou, Jianxin Wu, Chen-Lin Zhang, and Zhi-Hua Zhou. Minimal gated unit for recurrent neural networks. *International Journal of Automation and Computing*, 13(3):226–234, 2016.

A Derivation of the Attention Jacobian

In this appendix, we present the formal derivation of the linear operator $\mathcal{J}_{\text{att}}^{(l)}$ introduced in Section 4.1.2. The analysis is founded on matrix differential calculus, establishing the sensitivity relationship between the perturbations of the block state BS_{l-1} and the resulting attention matrix.

A.1 Definition of the Softmax Jacobian ($\mathcal{J}_{\text{softmax}}$)

Let $R \in \mathbb{R}^{\omega \times \omega}$ be the scaled *logits* matrix, defined as $R = \frac{QK^\top}{\sqrt{d_k}}$. We denote by $r_i^\top \in \mathbb{R}^\omega$ the i-th row of R, treated as a column vector for the purposes of vector differentiation.

We define the smoothed attention matrix $S = \text{softmax}(R)$, whose operation is applied independently over each row:

$$s_i = \text{softmax}(r_i) \in \mathbb{R}^\omega$$

Whose scalar components are given by:

$$s_{ij} = \frac{\exp(r_{ij})}{\sum_{t=1}^{\omega} \exp(r_{it})}, \quad i, j \in \{1, \ldots, \omega\}$$

A.1.1 Local derivative and Jacobian matrix

For a fixed row i, the partial derivative of the output with respect to the input is given by the symmetric structure:

$$\frac{\partial s_{ij}}{\partial r_{ik}} = s_{ij}(\delta_{jk} - s_{ik})$$

Where δ_{jk} represents the Kronecker delta. In matrix notation, this forms the local Jacobian matrix $J_{\text{sm}}(r_i) \in \mathbb{R}^{\omega \times \omega}$:

$$J_{\text{sm}}(r_i) = \text{diag}(s_i) - s_i s_i^\top$$

Consequently, for an infinitesimal perturbation of the input dr_i, the variation in the output is:

$$ds_i = J_{\text{sm}}(r_i)\, dr_i$$

A.1.2 Row-by-row matrix operator

We generalize the previous concept by defining the operator $\mathcal{J}^{(l)}_{\text{softmax}}$ that acts on the complete matrix of differentials $dR \in \mathbb{R}^{\omega \times \omega}$. The transformation is applied by applying the corresponding local Jacobian to each row:

$$dS = \mathcal{J}^{(l)}_{\text{softmax}}[dR] \implies (dS)^\top_{i,:} = J_{\text{sm}}(r_i)(dR)^\top_{i,:}, \quad \forall i \in \{1, \ldots, \omega\}$$

A.2 Derivation of the Attention Jacobian ($\mathcal{J}_{\text{att}}$)

We proceed to derive the sensitivity of the final attention matrix A with respect to the input state BS_{l-1}. We define the linear projections (following the notation $Inputs \times Weights$):

$$Q = BS_{l-1}W_q^{(l)}, \quad K = BS_{l-1}W_k^{(l)}, \quad V = BS_{l-1}W_v^{(l)}$$

And the composite operations:

$$R = \frac{QK^\top}{\sqrt{d_k}}, \quad S = \text{softmax}(R), \quad A = SV$$

Applying the product rule for matrix differentials on A, we obtain:

$$dA = (dS)V + S(dV) \tag{43}$$

We break down the differential terms as a function of dBS_{l-1}:

1. Values Differential (dV):

$$dV = dBS_{l-1}W_v^{(l)}$$

2. Logits Differential (dR): Applying the product rule on R:

$$dR = \frac{1}{\sqrt{d_k}}\left((dQ)K^\top + Q(dK)^\top\right)$$

Substituting the linear projections:

$$dR = \frac{1}{\sqrt{d_k}}\left((dBS_{l-1}W_q^{(l)})K^\top + Q(dBS_{l-1}W_k^{(l)})^\top\right)$$

3. Probabilities Differential (dS): Using the operator defined in Section A.1:

$$dS = \mathcal{J}^{(l)}_{\text{softmax}}[dR]$$

Finally, we substitute dS and dV into Equation 43, factoring dBS_{l-1} implicitly within the linear operators. We define the Attention Jacobian operator as the transformation:

$$\mathcal{J}^{(l)}_{\text{att}} : dBS_{l-1} \in \mathbb{R}^{\omega \times d_l} \mapsto dA_l \in \mathbb{R}^{\omega \times d_l}$$

Demonstrating that its explicit form is:

$$\mathcal{J}^{(l)}_{\text{att}}[dBS_{l-1}] = \underbrace{\left(\frac{1}{\sqrt{d_k}} \mathcal{J}^{(l)}_{\text{softmax}} \left[(dBS_{l-1} W_q^{(l)}) K^\top + Q (dBS_{l-1} W_k^{(l)})^\top \right] \right) V}_{\text{Scores Gradient Pathway}} + \underbrace{S(dBS_{l-1} W_v^{(l)})}_{\text{Values Gradient Pathway}}$$