SELECTIVE ROTARY POSITION EMBEDDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Position information is essential for language modeling. In softmax transformers, Rotary Position Embeddings (*RoPE*) encode positions through *fixed-angle* rotations, while in linear transformers, order is handled via input-dependent (selective) gating that decays historical information. Selectivity has generally been shown to improve language related tasks. Inspired by this, we introduce **Selective RoPE**, an *input-dependent* rotary embedding mechanism, that generalizes *RoPE*, and enables rotation in all angels for both linear and softmax transformers. We show that softmax attention already performs a hidden form of these rotations on query-key pairs, uncovering an implicit positional structure. We further show that in state-space models and gated linear transformers, the real part manages forgetting while the imaginary part encodes positions through rotations. We validate our method by equipping gated linear attention (GLA) with **Selective RoPE**, demonstrating that its input-dependent rotations improve performance in language modeling and on difficult sequence tasks like copying, state tracking, and retrieval.

1 INTRODUCTION

Transformers with softmax attention (Vaswani et al., 2017) form the backbone of modern Large Language Models (LLMs). Since attention is permutation-equivariant with respect to the input token order, positional information must be injected explicitly. Among exiting methods, Rotary Position Embedding (*RoPE*) has emerged as the popular choice due to its drop-in simplicity and strong empirical performance (Su et al., 2021). *RoPE* encodes positions by applying fixed-frequency rotations to queries and keys.

In parallel, efficient sequence models such as Linear Transformers and State Space Models (SSMs) (Katharopoulos et al., 2020; Yang et al., 2023; 2024b) have introduced selective gating as a mechanism to regulate the memory. Selectivity enables models to forget past information at input-dependent manner, greatly improving performance in language tasks and long-range reasoning. However, these models often only rely on forgetting alone, without an explicit mechanism for encoding rotations that capture relative positions (Gu & Dao, 2024; Dao & Gu, 2024).

The gap between these two types of positional encodings motivates our work. We introduce *Selective RoPE*, an input-dependent extension of *RoPE* that performs rotations at arbitrary, selective frequencies. *Selective RoPE* generalizes *RoPE* beyond fixed frequencies, while retaining its drop-in simplicity by directly rotating queries and keys. Intriguingly, we show that softmax attention already performs a hidden form of selective rotation via the perspective of Random Fourier Features (RFF).

We also analyze the roles of forget gates and rotations in gated linear transformers: the "*forget gate*," so named since it discards history, fails to encode rotary positional information, while rotation encodes positions but cannot decay the past. Our theoretical analysis provides a deeper understanding of rotary embeddings, followed by the perspective introduced in Barbero et al. (2025).

We show that diagonal linear and softmax transformers are enhanced by combining both mechanisms: *selective rotation* and the decaying *forget gate*. Conceptually and empirically, we show these gates play complementary roles, and equipping GLA and Transformer with *Selective RoPE* unlocks key capabilities such as state tracking, copying, and retrieval bottlenecks for these models.

Our contributions are as follows:

- ***Selective RoPE:*** We introduce an input-dependent rotary position embedding that generalizes RoPE from fixed to arbitrary, learnable rotations.

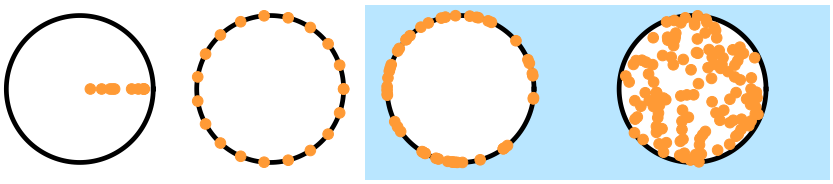

Figure 1: Our methods (right two columns) are with the light background (*cf.,* Table 1). **Left to right:** GLA, RoPE, *Selective RoPE* (ours), *Selective RoPE* + Decay (ours). As we observe, the forget gate only encodes positional information through scale. On the other hand, both RoPE and *Selective RoPE* allow for positional information to be encoded through rotation, with the selective variant taking advantage of arbitrary angles. Combining the two methods yields the best result.

- **Theoretical insight:** We show that softmax attention without explicit positional encoding implicitly performs random selective rotations via a Random Fourier Features expansion. We also find a suitable distribution for the temperatures on the selective frequencies that minimizes the error for this expansion.
- **Unification of decay and rotation:** We decompose the roles of imaginary and real components in gated linear transformers and diagonal SSMs: the imaginary part encodes positions via rotation, connecting the model to softmax attention, while the real part captures selective forgetting. A visualization of this relationship is available in Figure 1.
- **Empirical evidence:** We demonstrate empirical gains with *Selective RoPE* on linear attention for challenging sequence tasks, such as copying, state tracking and retrieval, while providing evidence for better state-tracking in softmax attention with *Selective RoPE*.

2 Background

2.1 Gated Linear Transformers

Standard causal softmax attention (Vaswani et al., 2017) projects a sequence of L inputs $(\boldsymbol{x}_t)_{t=1}^L$ into the sequence of outputs $(\boldsymbol{o}_t)_{t=1}^L$, where $\boldsymbol{x}_t, \boldsymbol{o}_t \in \mathbb{R}^d$:

$$\boldsymbol{q}_t, \boldsymbol{k}_t, \boldsymbol{v}_t = \boldsymbol{W}_q\boldsymbol{x}_t, \boldsymbol{W}_k\boldsymbol{x}_t, \boldsymbol{W}_v\boldsymbol{x}_t, \qquad \boldsymbol{o}_t = \frac{\sum_{\tau=1}^t \exp(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau)\boldsymbol{v}_\tau}{\sum_{\tau=1}^t \exp(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau)}. \tag{1}$$

with $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v \in \mathbb{R}^{d\times d}$ being the projection matrices.

Linear attention (Katharopoulos et al., 2020) replaces the exponential kernel $\exp(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau)$ in equation 1 with a kernel with a positive feature map $\phi(.) : \mathbb{R}^d \to (\mathbb{R}^+)^d$, which makes linear attention equivalent to an RNN:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1} + \boldsymbol{v}_t\phi(\boldsymbol{k}_t)^\top, \quad \boldsymbol{z}_t = \boldsymbol{z}_{t-1} + \phi(\boldsymbol{k}_t), \quad \boldsymbol{o}_t = \frac{\boldsymbol{S}_t\phi(\boldsymbol{q}_t)}{\boldsymbol{z}_t^\top\phi(\boldsymbol{q}_t)} = \frac{\sum_{\tau=1}^t \phi(\boldsymbol{q}_t)^\top\phi(\boldsymbol{k}_\tau)\boldsymbol{v}_\tau}{\sum_{\tau=1}^t \phi(\boldsymbol{q}_t)^\top\phi(\boldsymbol{k}_\tau)}. \tag{2}$$

Here $\boldsymbol{S}_t \in \mathbb{R}^{d\times d}$ and $\boldsymbol{z}_t \in \mathbb{R}^d$ are hidden states updated by the above recurrence. Moving forward, we subsume the feature map $\phi(.)$ into query-key vectors to reduce notation clutter. To control the influence of $\boldsymbol{S}_{t-1}$ and prevent it from dominating, equation 2 was enhanced with gating:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1}\boldsymbol{A}_t + \boldsymbol{v}_t\boldsymbol{k}_t^\top, \quad \boldsymbol{o}_t = \boldsymbol{S}_t\boldsymbol{q}_t. \tag{3}$$

where $\boldsymbol{A}_t$ is known as forget gate, which *selectively* discards historical information. Moreover, the normalization state $\boldsymbol{z}_t$ is typically omitted in gated linear transformers and SSMs such as *Mamba* (Gu & Dao, 2024) and *GLA* (Yang et al., 2023). Gated linear transformers construct the forget gate in different forms: *Mamba-2* (Gu & Dao, 2024) uses a scalar gate ($\boldsymbol{A}_t \in \mathbb{R}$); *GLA* and *Mamba* (Yang et al., 2023; Gu & Dao, 2024) adopt vector gates ($\boldsymbol{A}_t \in \mathbb{R}^d$); and *DeltaNet* (Yang et al., 2024b) employs a matrix gate via the Delta Rule ($\boldsymbol{A}_t = \boldsymbol{I} - \beta_t\boldsymbol{k}_t\boldsymbol{k}_t^\top$).

The forget gate $\boldsymbol{A}_t$ can also be interpreted as a position encoding, as it regulates the influence of past tokens on the current output t according to their *relative* distance (Yang et al., 2025b).

2.2 *RoPE* AND COMPLEX LINEAR ATTENTION

Unlike gated linear transformers, softmax attention does not encode token positions, and is therefore, referred to as *NoPE*, i.e., no position encoding (Kazemnejad et al., 2023). Rotary Position Embeddings (*RoPE*) address this by injecting relative position information through rotations of query-key pairs (Su et al., 2021).

For 2-dimensional queries and keys $\boldsymbol{q}_t, \boldsymbol{k}_\tau \in \mathbb{R}^2$, *RoPE* applies relative positional encoding through rotation matrix $\boldsymbol{R}_\omega$, which changes the exponential kernel in softmax attention as:

$$\exp\left(\boldsymbol{k}_\tau^\top \boldsymbol{R}_\omega^{t-\tau} \boldsymbol{q}_t\right) = \exp\left((\boldsymbol{R}_\omega^\tau \boldsymbol{k}_\tau)^\top (\boldsymbol{R}_\omega^t \boldsymbol{q}_t)\right), \quad \boldsymbol{R}_\omega = \begin{bmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{bmatrix}. \tag{4}$$

Here, ω determines the rotation frequency. Thus, the query at time t and key at time τ are rotated by $\boldsymbol{R}_\omega$, and the identity $(\boldsymbol{R}_\omega)^t = \boldsymbol{R}_{t\omega}$ shows that exponentiation by t corresponds to a rotation of angle $t\omega$. For d-dimensional queries and keys, $\boldsymbol{q}_t, \boldsymbol{k}_\tau \in \mathbb{R}^d$ are split into $d/2$ two-dimensional subvectors, each rotated independently with its own frequency. This yields a block-diagonal rotation matrix $\boldsymbol{R} \in \mathbb{R}^{d\times d}$ defined as:

$$\boldsymbol{R} = \bigoplus_{n=1}^{d/2} \boldsymbol{R}_{\omega_n}, \tag{5}$$

where each $\boldsymbol{R}_{\omega_k} \in \mathbb{R}^{2\times 2}$ is a rotation matrix parameterized by a frequency ω_k, and $\oplus$ denotes the concatenation of matrices along the diagonal. Then, the exponential kernel in softmax for d-dimensional queries and keys becomes:

$$\exp\left(\boldsymbol{k}_\tau^\top \boldsymbol{R}^{t-\tau} \boldsymbol{q}_t\right) = \exp(\boldsymbol{k}_\tau^\top \boldsymbol{R}^t (\boldsymbol{R}^\tau)^{-1} \boldsymbol{q}_t) = \exp\left((\boldsymbol{R}^\tau \boldsymbol{k}_\tau)^\top (\boldsymbol{R}^t \boldsymbol{q}_t)\right), \tag{6}$$

which encodes relative positions through coordinated rotation across all 2D frequency subspaces. *A key connection is that RoPE is identical to a linear transformer with imaginary gate.* Let $\tilde{\boldsymbol{q}}_t, \tilde{\boldsymbol{k}}_t \in \mathbb{C}^{\frac{d}{2}}$ be the complex queries and keys with real and imaginary parts as:

$$\tilde{\boldsymbol{q}}_t = \tilde{\boldsymbol{q}}_t^R + i\tilde{\boldsymbol{q}}_t^I, \quad \tilde{\boldsymbol{k}}_t = \tilde{\boldsymbol{k}}_t^R + i\tilde{\boldsymbol{k}}_t^I$$

Let $\bar{\boldsymbol{R}} = \exp(i\boldsymbol{\Omega}) = \mathrm{diag}(e^{i\omega_1}, \ldots, e^{i\omega_{d/2}}) \in \mathbb{C}^{\frac{d}{2}\times\frac{d}{2}}$ denote a diagonal, purely imaginary forget gate, with the corresponding recurrence:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1}\bar{\boldsymbol{R}} + \boldsymbol{v}_t \tilde{\boldsymbol{k}}_t^\top, \quad \boldsymbol{o}_t = \mathcal{R}\{\boldsymbol{S}_t \tilde{\boldsymbol{q}}_t\}, \tag{7}$$

where $\mathcal{R}\{\cdot\}$ denotes taking the real part. This recurrence is equivalent to applying *RoPE* to a linear transformer with real-valued queries and keys $\boldsymbol{q}_t, \boldsymbol{k}_t \in \mathbb{R}^d$, where the real and imaginary parts of each complex dimension are interleaved (Su et al., 2021):

$$\boldsymbol{q}_t = \bigoplus_{n=1}^{d/2} \begin{bmatrix} \tilde{\boldsymbol{q}}_{t,n}^R \\ \tilde{\boldsymbol{q}}_{t,n}^I \end{bmatrix}, \quad \boldsymbol{k}_t = \bigoplus_{n=1}^{d/2} \begin{bmatrix} \tilde{\boldsymbol{k}}_{t,n}^R \\ \tilde{\boldsymbol{k}}_{t,n}^I \end{bmatrix}.$$

In practice, *RoPE* is implemented in the real domain by using equation 6, and rotating keys ($\boldsymbol{k}_\tau^{\text{pe}} = \boldsymbol{R}^\tau \boldsymbol{k}_\tau$) and queries ($\boldsymbol{q}_t^{\text{pe}} = \boldsymbol{R}^t \boldsymbol{q}_t$) eliminating the need for complex-valued operations (aka *RoPE* trick). Specifically, simply through the following operation:

$$\boldsymbol{q}_t^{\text{pe}} = \begin{bmatrix} \boldsymbol{q}_{t,0} \cdot \cos(\omega_0) - \boldsymbol{q}_{t,1} \cdot \sin(\omega_0) \\ \boldsymbol{q}_{t,1} \cdot \cos(\omega_0) + \boldsymbol{q}_{t,0} \cdot \sin(\omega_0) \\ \vdots \\ \boldsymbol{q}_{t,d-1} \cdot \cos(\omega_{d/2}) - \boldsymbol{q}_{t,d} \cdot \sin(\omega_{d/2}) \\ \boldsymbol{q}_{t,d} \cdot \cos(\omega_{d/2}) + \boldsymbol{q}_{t,d-1} \cdot \sin(\omega_{d/2}) \end{bmatrix}, \boldsymbol{k}_\tau^{\text{pe}} = \begin{bmatrix} \boldsymbol{k}_{\tau,0} \cdot \cos(\omega_0) - \boldsymbol{k}_{\tau,1} \cdot \sin(\omega_0) \\ \boldsymbol{k}_{\tau,1} \cdot \cos(\omega_0) + \boldsymbol{k}_{\tau,0} \cdot \sin(\omega_0) \\ \vdots \\ \boldsymbol{k}_{\tau,d-1} \cdot \cos(\omega_{d/2}) - \boldsymbol{k}_{\tau,d} \cdot \sin(\omega_{d/2}) \\ \boldsymbol{k}_{\tau,d} \cdot \cos(\omega_{d/2}) + \boldsymbol{k}_{\tau,d-1} \cdot \sin(\omega_{d/2}) \end{bmatrix},$$

where $\boldsymbol{q}_{t,j}$ and $\boldsymbol{k}_{\tau,j}$ denote the j^{th} elements of $\boldsymbol{q}_t$ and $\boldsymbol{k}_\tau$, respectively.

3 THE IMPLICIT ROTATION IN SOFTMAX ATTENTION

In this section, we analyze softmax attention through Random Fourier Feature (RFF) approximation, showing that it implicitly applies random rotations. We then derive the optimal temperature that minimizes this approximation error.

Gate Type: Gate Formulation	Selectivity	Model Examples	Gate Spectrum
Decay : $\boldsymbol{A}_t = \gamma$	✗	RetNet	
Decay : $\boldsymbol{A}_t = \sigma(\boldsymbol{W}\boldsymbol{x}_t)$	✔	Mamba, Mamba2, GLA, HGRN2, RWKV6	
Rotation : $\boldsymbol{A}_t = \exp(i\boldsymbol{\Omega})$	✗	Transformer+*RoPE*	
Decay+Rotation: $\boldsymbol{A}_t = \sigma(\boldsymbol{W}\boldsymbol{x}_t) \cdot \exp(i\boldsymbol{\Omega})$	✔	FoX+*RoPE*	
Rotation : $\boldsymbol{A}_t = \exp(i\boldsymbol{\Omega}\boldsymbol{q}_t)$	✔	*Selective RoPE*	
Decay+Rotation: $\boldsymbol{A}_t = \sigma(\boldsymbol{W}\boldsymbol{x}_t) \cdot \exp(i\boldsymbol{\Omega}\boldsymbol{q}_t)$	✔	*Selective RoPE*+GLA	

Table 1: Comparison of different Transformers and their corresponding forget gates. Dots indicate the relative position of two query-key pairs on the unit circle, representing their encoded distance.

3.1 Random Rotations in Softmax Attention

We start with the connection between RFFs and the softmax attention (Peng et al., 2021), and illustrate that rotation is an integral component in softmax attention. Specifically, starting from the definition of the softmax attention:

$$\boldsymbol{o}_t = \frac{\boldsymbol{s}_t}{\boldsymbol{z}_t}, \quad \boldsymbol{s}_t = \sum_{\tau=1}^{t} \exp\Big(\tfrac{1}{\sqrt{d}}\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\Big) \cdot \boldsymbol{v}_\tau, \quad \boldsymbol{z}_t = \sum_{\tau=1}^{t} \exp\Big(\tfrac{1}{\sqrt{d}}\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\Big),$$

with $\boldsymbol{q}_t, \boldsymbol{k}_\tau \in \mathbb{R}^d$. For simplicity, we omit the softmax temperature $1/\sqrt{d}$. We define the RFF kernel as $\phi_{\boldsymbol{\omega}}(\boldsymbol{x}) = \exp\Big(\frac{\|\boldsymbol{x}\|_2^2}{2} + i\boldsymbol{\omega}^\top \boldsymbol{x}\Big)$, the real component of the expectation of which is:

$$\mathcal{R}\left\{\mathbb{E}_{\boldsymbol{\omega}\sim\mathcal{N}(0,\boldsymbol{I})}\left[\phi_{\boldsymbol{\omega}}(\boldsymbol{q}_t)\phi_{\boldsymbol{\omega}}(\boldsymbol{k}_\tau)\right]\right\} = \exp\big(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\big). \tag{8}$$

By the law of large numbers, with $\boldsymbol{\omega}_j \sim \mathcal{N}\left(0, \sigma^2\boldsymbol{I}\right)$ for $j \in \{1, 2, \cdots, D\}$ and $\sigma = 1$ we can write: $\boldsymbol{s}_t = \lim_{D\to\infty} \mathcal{R}\left\{\frac{1}{D}\sum_{j=1}^{D} \hat{\boldsymbol{s}}_{t,j}\right\}$, where we define:

$$\hat{\boldsymbol{s}}_{t,j} = \sum_{\tau=1}^{t} \phi_{\boldsymbol{\omega}_j}\left(\boldsymbol{q}_t\right) \cdot \phi_{\boldsymbol{\omega}_j}\left(\boldsymbol{k}_\tau\right) \cdot \boldsymbol{v}_\tau.$$

Let us define $\hat{\boldsymbol{S}}_t^D \in \mathbb{R}^{d\times D}$ as the matrix with its j^{th} column equal to $\hat{\boldsymbol{s}}_{t,j}$. Following the definition of $\phi_\omega(.)$, we can write $\hat{\boldsymbol{s}}_{t,j}^D$ in a recurrent form as:

$$\hat{\boldsymbol{s}}_{t,j} = \exp\Big(\tfrac{\|\boldsymbol{q}_t\|_2^2 - \|\boldsymbol{q}_{t-1}\|_2^2}{2}\Big) \exp\big(i\boldsymbol{\omega}_j^\top(\boldsymbol{q}_t - \boldsymbol{q}_{t-1})\big)\, \hat{\boldsymbol{s}}_{t-1,j} + \phi_{\boldsymbol{\omega}_j}(\boldsymbol{q}_t) \cdot \phi_{\boldsymbol{\omega}_j}(\boldsymbol{k}_t) \cdot \boldsymbol{v}_t. \tag{9}$$

Note that in (9) the term $\exp\Big(\frac{\|\boldsymbol{q}_t\|_2^2 - \|\boldsymbol{q}_{t-1}\|_2^2}{2}\Big)$ can introduce instability to the recurrence. Therefore, following the standard in both linear transformers (Yang et al., 2024b; 2023; Yang et al.; Lin et al., 2025) and deep softmax transformers (Henry et al., 2020), we assume L_2 normalization over the query and key. Defining $\phi(x)$ as the vector with its j^{th} element equal to $\phi_{\boldsymbol{\omega}_j}(x)$, we write the recurrence over $\hat{\boldsymbol{S}}_t$ as:

$$\hat{\boldsymbol{S}}_t^D = \hat{\boldsymbol{S}}_{t-1}^D \bar{\boldsymbol{R}}_t + \boldsymbol{v}_t \tilde{\boldsymbol{k}}_t^\top, \quad \bar{\boldsymbol{R}}_t = \exp\big(i\boldsymbol{\Omega}(\boldsymbol{q}_t - \boldsymbol{q}_{t-1})\big), \quad \tilde{\boldsymbol{k}}_t = \phi(\boldsymbol{q}_t) \circ \phi(\boldsymbol{k}_\tau), \tag{10}$$

where $\circ$ corresponds to the Hadamard product of two vectors.

Similarly defining approximation of $\hat{\boldsymbol{Z}}_t^D \in \mathbb{R}^{1\times D}$ via RFF, we arrive at the following approximation of the softmax attention as a complex-valued linear attention: $\boldsymbol{o}_t = \lim_{D\to\infty} \frac{\mathcal{R}\{\hat{\boldsymbol{S}}_t^D \mathbf{1}\}}{\mathcal{R}\{\hat{\boldsymbol{Z}}_t^D \mathbf{1}\}}$. For a detailed derivation, please refer to App. A.2.

In conclusion, softmax attention implicitly is applying selective rotations through the gate $\bar{\boldsymbol{R}}_t$

3.2 Optimal variance for Fourier features

The equivalence of the RFF kernel in equation 8 with $\sigma = 1$ holds only in expectation. For a limited number of samples (D), we instead choose the variance of the RFFs carefully in our complex linear attention model equation 10. We obtain the optimal variance for a given query-key pair as follows

Theorem 1 *Let the expected error of the RFF kernel over $\boldsymbol{\omega}_j \sim \mathcal{N}\left(0, \sigma^2 \boldsymbol{I}\right)$ be as follows: $ERR\left[\boldsymbol{q}_t, \boldsymbol{k}_\tau\right] = \mathbb{E}_{\boldsymbol{\omega}_j}\left[\left(\frac{1}{D}\sum_{j=1}^{D} \phi_{\boldsymbol{\omega}_j}(\boldsymbol{q}_t) \cdot \phi_{\boldsymbol{\omega}_j}(\boldsymbol{k}_\tau) - \exp\left(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\right)\right)^2\right]$. Then, for a given a pair of L_2 normalized query and key, the optimal value of σ is equal to $\sigma = \tan\left(\frac{\arccos\left(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\right)}{2}\right)$.*

Theorem 1 provides the optimal variance for RFFs for a single query-key pair. Extending this, we define the rotation matrix as $\hat{\boldsymbol{R}}_t = \exp(i\boldsymbol{\Omega}\Theta(\boldsymbol{q}_t - \boldsymbol{q}_{t-1}))$, where Θ is a diagonal matrix of temperatures. Assuming the angle between the queries and keys are uniformly distributed in $[0, 2\pi]$, the optimal temperatures follow $\tan^2(\frac{\theta}{2})$ with $\theta \sim \mathcal{U}[0, 2\pi]$. Interestingly, this distribution closely resembles the exponentially decaying frequencies used in *RoPE*. Together, these observations lead to our first key insight:

Insight *1*): *Softmax attention implicitly applies a **selective rotation**, to encode relative positional information between tokens. Since $\|\boldsymbol{R}_t\|_2 = 1$, it preserves the norm of query-key pairs and does not forget past information. The optimal temperature for these rotations follows a exponentially decaying pattern.*

4 Selective *RoPE*: Rotation in all Angles

From section 3.1, we observe that there is a direct relationship between performing *selective rotations* in softmax attention thorough RFF recurrence style. This relationship emphasizes the importance of rotation as a way to *preserve positional information*, given the superior recall ability of softmax attention (Arora et al., 2024a; Jelassi et al., 2024). Following this intuition and the proven benefits of selectivity in linear transformers (Yang et al., 2023; Gu & Dao, 2024; Arora et al., 2024a; Dao & Gu, 2024), we introduce ***Selective RoPE*** as a generalization of *RoPE* with input-dependent rotations allowing arbitrary frequencies for softmax and linear attentions.

The construction of *Selective RoPE* is simple: rather than using fixed, pre-defined frequencies $\boldsymbol{\Omega}$, we make them input-dependent (i.e., selective). Inspired by Figure 2, the frequencies are computed as a function of the queries $\boldsymbol{q}_t$ through a linear projection of $\boldsymbol{W}_{\boldsymbol{\Omega}}$, which results in less parameters compared to extracting them from input $\boldsymbol{x}_t$. The definition of *Selective RoPE*, given both in its real-valued matrix form and its equivalent imaginary recurrence (analogous to *RoPE*'s recurrence in eq. (7)), is presented in section 4. *Moreover, Selectve RoPE shares the simplicity of integration of RoPE trick as shown in the pseudo code in Figure 2.* Inspired by the rotation gate in Yang et al. (2025b), we found it to be helpful to add an input-dependent gating mechanism to the phase of the rotation matrix, in order to endow the model with the ability to not perform rotation.

5 Gated Linear Transformers Equipped with *Selective RoPE*

In this section, we first analyze the roles of real and imaginary parts in non-selective SSMs (S4D), then show how *Selective RoPE* naturally arises from a linear transformer with a complex forget gate. Building on this, we demonstrate that rotation and norm decay play complementary roles, and that linear transformers unlock new capabilities when equipped with *Selective RoPE*.

5.1 Role of Real and Imaginary Parts in S4D

We start our analysis with non-selective diagonal SSMs and prove the distinct roles of the real and imaginary components. SSMs can be derived from continuous-time representations, expressed as:

$$\frac{d\boldsymbol{s}(t)}{dt} = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{B}\boldsymbol{x}(t), \quad \boldsymbol{o}(t) = \boldsymbol{C}^\top \boldsymbol{s}(t), \quad K(t) = \boldsymbol{C}^\top e^{\boldsymbol{A}t}\boldsymbol{B}, \quad \boldsymbol{o}(t) = K(t) * \boldsymbol{x}(t) \tag{13}$$

Selective RoPE:

$$\bar{\boldsymbol{R}}_t = \exp(i\boldsymbol{\Omega}_t), \text{(Imaginary)} \tag{11}$$

$$= \bigoplus_{n=1}^{d/2} \begin{bmatrix} \cos(\boldsymbol{\Omega}_{n,t}) & -\sin(\boldsymbol{\Omega}_{n,t}) \\ \sin(\boldsymbol{\Omega}_{n,t}) & \cos(\boldsymbol{\Omega}_{n,t}) \end{bmatrix}, \text{(Real)}$$

$$\boldsymbol{\Omega}_t = \text{Conv1D}(\boldsymbol{W}_{\boldsymbol{\Omega}}\tilde{\boldsymbol{q}}_t), \tag{12}$$

$$\boldsymbol{W}_{\boldsymbol{\Omega}} \in \mathbb{R}^{d \times d/2}$$

```
def Selective_RoPE(Q,K,W_Ω,θ, Temp):
    Ω = cumsum( Conv1d ( Temp* (W_Ω@Q) ) )
    return Apply_RoPE(Q,K,Ω)

def Apply_RoPE(Q,K,Ω) # RoPE trick
    Q_r[0::2] = Q[0::2]*cosΩ - Q[1::2]*sinΩ
    Q_r[1::2] = Q[1::2]*cosΩ + Q[0::2]*sinΩ
    K_r[0::2] = K[0::2]*cosΩ - K[1::2]*sinΩ
    K_r[1::2] = K[1::2]*cosΩ + K[0::2]*sinΩ
    return Q_r, K_r
```

Figure 2: ***Left)*** The formulation of *Selective RoPE*. The phase of the rotation matrix is the result of a linear projection, a channel-wise causal convolution, and a channel-wise multiplication with the temperatures. ***Right)*** Pseudo code of *Selective RoPE*. Thanks to the RoPE trick (Su et al., 2021), implementing *Selective RoPE* is possible in pure PyTorch. A detailed implmenetation is available at Figure 9.

In diagonal SSMs (S4D) with diagonal $\boldsymbol{A}$, S4D (Gu et al., 2022) initializes the imaginary part of the state matrix as $\boldsymbol{A}_n = i\omega_n$ ($n \in [0, N]$, roots of unity), from which the output is derived as:

$$\boldsymbol{o}(t) = \sum_{n=1}^{N} \boldsymbol{B}_n \boldsymbol{C}_n e^{i\omega_n t} \int_{-\infty}^{\infty} e^{-i\omega_n \tau} \boldsymbol{x}(\tau)\boldsymbol{u}_t(\tau)d\tau, \quad \boldsymbol{u}_t(\tau) = \begin{cases} 1, & 0 \le \tau \le t \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where $\boldsymbol{u}_t(\tau)$ is a step-window function. The integral above is equivalent to computing the Fourier Transform of the windowed signal $\boldsymbol{x}(\tau)\,\boldsymbol{u}_t(\tau)$ at frequency ω_n. Duality between convolution in the time domain and multiplication in the frequency domain simplifies the equation 14 to:

$$\boldsymbol{o}(t) = \sum_{n=1}^{N} \boldsymbol{B}_n \boldsymbol{C}_n (\boldsymbol{X}_{\omega_n} * \boldsymbol{U}_{t,\omega_n}), \quad \boldsymbol{U}_{t,2\omega} = \frac{\sin(\omega t)}{\omega} e^{-i\omega t} \tag{15}$$

with $\boldsymbol{X}_{\omega_n}$ and $\boldsymbol{U}_{t,\omega_n}$ denote the Fourier transforms of $\boldsymbol{x}(\tau)$ and $\boldsymbol{u}_t(\tau)$, respectively. The input spectrum $\boldsymbol{X}_\omega$ is convolved with the window spectrum $\boldsymbol{U}_{t,\omega}$, causing distortion, a phenomenon known as *spectral leakage*. In the discrete domain, the integral in equation 14 becomes a summation:

$$\boldsymbol{o}_t = \sum_{n=0}^{N} \boldsymbol{C}_n \boldsymbol{B}_n \sum_{\tau=0}^{t} \exp\left(-\tfrac{2\pi i n \tau}{N}\right) \boldsymbol{x}_\tau. \tag{16}$$

where $\omega_n = \frac{2\pi n i}{N}$ and $\Delta = \frac{1}{N}$. Thus, S4D with a purely imaginary state matrix $\boldsymbol{A}$ acts as a spectral analyzer: it accurately computes the N-point DFT of for $t \le N$, but for $t > N$, suffers from **spectral leakage** since the state size can at most represent N frequencies. Therefore, the higher frequencies are being aliased or overwritten.

Equation 16 motivates the need to prevent spectral leakage, typically addressed by windowing in signal processing. S4D implicitly achieves this by using a complex state matrix $\boldsymbol{A}$ with the real part acting as a ***window function***, a classical solution to spectral leakage (Oppenheim, 1999). Concretely, with $\boldsymbol{A} = \exp(-\alpha_n\Delta + 2\pi i n\Delta)$, S4D performs a windowed DFT using a *Poisson window* (Smith III, 2011), thereby avoiding spectral leakage. Its output can be written as:

$$\boldsymbol{o}_t = \sum_{n=0}^{N} \boldsymbol{C}_n \boldsymbol{B}_n \sum_{\tau=0}^{t} \exp\left(-\tfrac{2\pi i n \tau}{N}\right) \boldsymbol{x}_\tau \underbrace{\exp(-\alpha_n \Delta \tau)}_{\boldsymbol{w}_\tau}, \tag{17}$$

where $\boldsymbol{w}_\tau$ is the Poisson window and $\Delta = \frac{1}{N}$ is chosen for clarity in the DFT formulation. Thus, the real part of $\boldsymbol{A}$ in S4D acts as a window, suppressing spectral leakage and enabling undistorted spectral representations.

Summary of the Section: *In diagonal SSMs (S4D family), the two real and imaginary parts of state matrix $\boldsymbol{A}$ serve distinct but complementary roles: **Imaginary** parts extract spectral information, while **Real** parts suppress leakage and ensure clean representation of the spectrum.*

5.2 Rotation and Forget Gates

In this section, we extend the analogy from section 5 to linear transformers. Considering the general gated linear transformer's recurrence and its unrolled form:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1}\boldsymbol{A}_t + \boldsymbol{v}_t\tilde{\boldsymbol{k}}_t^\top, \quad \boldsymbol{o}_t = \mathcal{R}\{\boldsymbol{S}_t\tilde{\boldsymbol{q}}_t\}, \quad \boldsymbol{o}_t = \sum_{\tau=1}^{t} \boldsymbol{v}_\tau \, \mathcal{R}\left\{\tilde{\boldsymbol{k}}_\tau^\top \Big(\prod_{\kappa=\tau}^{t} \boldsymbol{A}_\kappa\Big)\tilde{\boldsymbol{q}}_t\right\}. \tag{18}$$

where $\tilde{\boldsymbol{k}}_t, \tilde{\boldsymbol{q}}_t \in \mathbb{C}^d$, $\boldsymbol{v}_t \in \mathbb{R}^d$, and the diagonal gate $\boldsymbol{A}_t \in \mathbb{C}^{d\times d}$. We adopt the complex formulation of linear transformer due to its connection with *RoPE* (section 2.2). In practice, several linear transformers parameterize the gate in log-space as $\boldsymbol{A}_t = \exp(-\boldsymbol{\Lambda}_t + i\boldsymbol{\Omega}_t)$, where $\boldsymbol{\Lambda}_t, \boldsymbol{\Omega}_t \in \mathbb{R}^{d\times d}$ are diagonal, learned during training and $\boldsymbol{\Lambda}_t$ is contractive for recurrence stability (Orvieto et al., 2023). Expanding the cumulative product over forget gate we achieve:

$$\begin{aligned}
\boldsymbol{o}_t &= \mathcal{R}\left\{\sum_{\tau=1}^{t}\left(\tilde{\boldsymbol{q}}_t\prod_{\kappa=1}^{t}\exp(i\boldsymbol{\Omega}_\kappa)\right)^\top\left(\prod_{\kappa=\tau+1}^{t}\exp(-\boldsymbol{\Lambda}_\kappa)\right)\left(\prod_{\kappa=1}^{\tau}\exp(-i\boldsymbol{\Omega}_\kappa)\tilde{\boldsymbol{k}}_\tau\right)\cdot\boldsymbol{v}_\tau\right\}, \\
&= \sum_{\tau=1}^{t}\boldsymbol{v}_\tau\sum_{n=1}^{d}\underbrace{|\tilde{\boldsymbol{k}}_{n,\tau}|\,|\tilde{\boldsymbol{q}}_{n,t}|\,e^{-\sum_{\kappa=\tau}^{t}\boldsymbol{\Lambda}_{n,\kappa}}}_{\text{Norm Decay}}\underbrace{\cos\Big(\angle\tilde{\boldsymbol{q}}_{n,t}+\angle\tilde{\boldsymbol{k}}_{n,\tau}+\sum_{\kappa=\tau}^{t}\boldsymbol{\Omega}_{n,\kappa}\Big)}_{\text{Rotation}}.
\end{aligned} \tag{19}$$

Here, $\tilde{\boldsymbol{q}}_{n,t}, \tilde{\boldsymbol{k}}_{n,\tau}, \boldsymbol{\Lambda}_{n,t}, \boldsymbol{\Omega}_{n,t} \in \mathbb{R}$ denote the n-th elements of the respective vectors or diagonal matrices at time t. *Equation (19) shows that the real part of $\boldsymbol{A}$ causes norm decay (forgetting), while the imaginary part rotates query-key angles (position encoding).* Together, they span the full unit disk of query-key relations, as visualized in table 1.

Similar to *RoPE*, selective rotations can be applied directly to queries and keys as in equation 19: $\boldsymbol{q}_t^{\text{pe}} = \tilde{\boldsymbol{q}}_t\prod_{\kappa=1}^{t}\exp(i\boldsymbol{\Omega}_\kappa)$, $\boldsymbol{k}_\tau^{\text{pe}} = \exp(-i\boldsymbol{\Omega}_\kappa)\tilde{\boldsymbol{k}}_\tau$, using the standard *RoPE* trick, via exact code presented in section 4. *Hence, the imaginary part of $\boldsymbol{A}_t$, $\exp(i\boldsymbol{\Omega}_t)$, is equivalent to Selective RoPE and can be implemented directly in practice.* This results in our second insight on applying *Selective RoPE* for gated linear transformers:

Insight *2):* *Linear Transformers can be enhanced by using both forgetting via real decay and rotation via imaginary gate. The rotation component enriches the connection between linear attention and softmax attention, providing better memory, while the decay component prevents spectral leakage and filters older information.*

5.3 Connections

In this section, we note the connection between our contribution and some of current literature - (Yang et al., 2024b; 2025b; Yang et al.; Lin et al., 2025). For selective rotation as a method of position embedding, we point to the **DeltaNet** method and the **PaTH** as two methods most closely connected to our work. In App. A.4, we show that *Selective RoPE* can be seen as a special variant of these models when used in linear attention or softmax attention, respectively. Similarly, considering the relationship between softmax attention and *Selective RoPE*, we can view the softmax attention with forget gate (**FoX**) as closely related to a GLA with *Selective RoPE*. In App. B.1, we further expand on this connection.

6 Experiments

We test the performance of *Selective RoPE* on synthetic and real world language modeling tasks. For this, we implement *Selective RoPE* in `flash-linear-attention` (Yang & Zhang, 2024) and use the model implementations provided in the library. We mainly compare *Selective RoPE* to NoPE and RoPE applied to Gated Linear Attention (GLA) (Yang et al., 2024a).

6.1 Synthetic Language Tasks

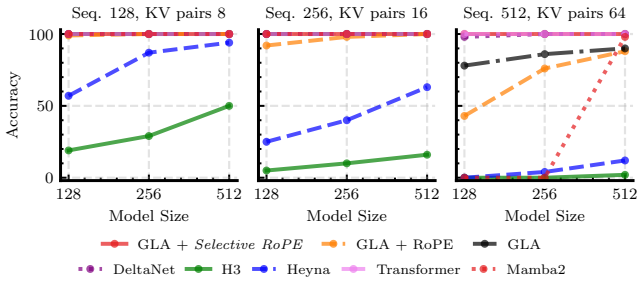

Figure 3: Multi-Query Associative Recall (MQAR) of GLA with *Selective RoPE* compared to RoPE, NoPE and other architectures. MQAR feeds a sequence of key-value pairs followed by several queries and asks the model to output the corresponding values in order. KV = key-value.

Table 2: MAD benchmark results.

Model	Compress	Fuzzy Recall	In-Context Recall	Memorize	Noisy Recall	Selective Copy	Average
GLA							
NoPE	77.1	8.8	92.7	**30.2**	94.0	93.1	66.0
RoPE	77.0	8.8	92.5	29.7	88.1	93.6	64.5
Selective RoPE	**78.3**	**9.1**	**97.9**	28.3	**95.5**	**97.3**	**67.7**

To investigate which capabilities of linear attention are improved when using *Selective RoPE*, we run experiments on synthetic tasks. For this, we mostly focus on recall, since it is essential for language modeling (Arora et al., 2024a;b) and a good proxy for performance at scale.

MQAR. We evaluate GLA + *Selective RoPE* on Multi-Query Associative Recall, following the same experimental setup as in Arora et al. (2024a, Figure 2) with a finer learning rate grid, as this has been show to improve performance (Okpekpe & Orvieto, 2025) (cf. Appendix E.2). The results are shown in Figure 3. While GLA with RoPE and NoPE is not able to achieve perfect accuracy at the longest sequence length, GLA with *Selective RoPE* achieves 100% at all considered sequence lengths and number of key-value pairs.

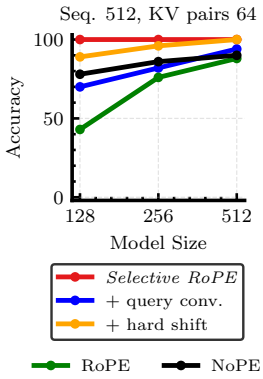

Figure 4: MQAR ablations.

Contrary to our results on language modeling, we observed that the causal 1D-convolution and hard-shift (cf. Section 3.1) both hurt performance and *Selective RoPE* performs best on MQAR with just the phase projection from the queries as shown in Figure 4. We hypothesize that this is a learnability rather than a capability issue, since the model could also learn to ignore past queries in the causal 1D-convolution and place all weight on the current query $\boldsymbol{q}_t$.

MAD and Copying. We also evaluate our method on the MAD benchmark suite (Poli et al., 2024) which tests a models ability to store and recall information within its context. Here, we note that using *Selective RoPE* consistently improves performance over NoPE and RoPE on almost all considered tasks. We also evaluate string copying following Jelassi et al. (2024). This task differs from *Selective Copy* in MAD in that the entire input sequence has to be copied token-by-token after the models is presented with a `<copy>` token. The results in Figure 5 show that *Selective RoPE* again improves over the alternatives and learns to length extrapolate very robustly. The poor result of RoPE is reported in prior works (Jelassi et al., 2024; Li et al., 2024) and attributable to it's generally poor length extrapolation performance without fine-tuning on longer sequence lengths.

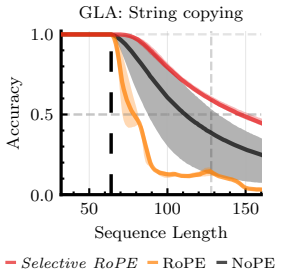

Figure 5: Copying accuracy of GLA with CIs. Dashed line is the training sequence length.

State Tracking. A common way to evaluate the expressivity of a model is *state tracking* on permutation composition (Liu et al., 2023). Recently, it has been shown, that SSMs and linear RNNs are not even capable of learning parity (Merrill et al., 2024), which amounts to permutation composition on the symmetric group of two elements, S_2, and that one needs to extend the eigenvalue range of the state transition $\boldsymbol{A}_t$ from $[0, 1]$ to $[-1, 1]$ (Grazzi et al., 2025). In Figure 6 we see that

Model	Wiki. ppl ↓	LMB. ppl ↓	LMB. acc ↑	PIQA acc ↑	Hella. acc_n ↑	Wino. acc ↑	ARC-e acc ↑	ARC-c acc_n ↑	Avg.
NoPE	28.24	37.37	32.1	64.8	34.3	52.2	45.8	23.5	46.05
Selective RoPE	**23.21**	**21.82**	**38.1**	**70.8**	**48**	**54**	**53.4**	**24.7**	**48.16**

Table 3: Evaluation results of 370M on tasks from `lm-eval-harness` (Gao et al., 2024).

GLA with *Selective RoPE* is able to learn and length, extrapolate on S_2. This is in line with our expectations since the input dependent rotations allow it to model "flips" depending on the input either being a 0 or a 1, while GLA with NoPE and RoPE does not even learn the training context length. This places GLA + *Selective RoPE* outside the TC^0 complexity class (Merrill et al., 2024). Similarly, we can see that *Selective RoPE* also improves the state tracking abilities in Transformers (i.e., softmax attention) allowing them to solve the parity problem up to, and slightly more, than the train sequence length. To the best of our knowledge, Transformer with *Selective RoPE* is the only variant of Transformers capable of solving the parity task with a single layer up to this sequence length (Liu et al., 2023). We also experiment on A_3 with a 2-layer DeltaNet Yang et al. (2024b), which is the permutation composition on the symmetric group of three elements, limited to even permutations. As we can observe, *Selective RoPE* improves the expressivity of the model up to a point where it is capable of solving A_3 up to the training sequence length. To the best of our knowledge, this is the first time these results have been presented for our choice of model on this task.

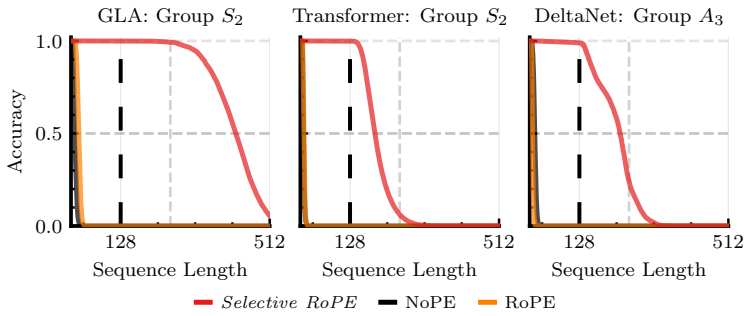

Figure 6: State tracking peformance of GLA, Transformer, and DeltaNet with different positional embeddings on S_2 and A_3. GLA and Transformer were trained with *one* layer whereas DeltaNet was trained with *two* layers. Vertical dashed line indicates training sequence length.

6.2 Language Modeling

For our language modeling experiments we follow the same training setup as in Siems et al. (2025a) and train a 370M GLA (Yang et al., 2024a). Our main experiments are carried out at 35B tokens ($\approx 5\times$ Chinchilla). To account for differences optimal learning rates for the considered positional embedding schemes, we sweep learning rates following Orvieto & Gower (2025) at the largest scale and early stop under-performing runs due to compute constraints. The downstream evaluation performance of GLA with *Selective RoPE* is shown in Table 3. During training, we observed loss spikes when using *Selective RoPE* which we were able to mitigate by placing a sigmoid gate on the phase (angle) projection. This also allows us to not rotate, similarly to PaTH (Yang et al., 2025a) (*cf.* Section 4). As shown in Table 3, using *Selective RoPE* consistently improves over NoPE.

7 Conclusion

We introduced *Selective RoPE*, an input-dependent rotary position embedding that generalizes RoPE from fixed to arbitrary, learnable rotations. Our theory shows (i) softmax attention admits a complex linear formulation that implicitly performs *selective rotations*, and (ii) this complex formulation introduces spectral leakage, which can be supressed through the forget gate mechanism. Empirically, equipping GLA with *Selective RoPE* improves recall-centric synthetic tasks and strengthens language modeling downstream performance .

REFERENCES

Niccol Ajroldi. plainlm: Language model pretraining in pytorch. https://github.com/Niccolo-Ajroldi/plainLM, 2024.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and improving recall in efficient language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024a.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. Simple linear attention language models balance the recall-throughput tradeoff. 2024b. URL https://openreview.net/forum?id=e93ffDcpH3.

F. Barbero, A. Vitvitskyi, C. Perivolaropoulos, R. Pascanu, and P. Veličković. Round and round we go! what makes rotary positional encodings useful? In *The Thirteenth International Conference on Learning Representations (ICLR'25)*. ICLR, 2025.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In *Forty-first International Conference on Machine Learning*, 2024.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

R. Grazzi, J. Siems, A. Zela, J. Franke, F. Hutter, and M. Pontil. Unlocking State-Tracking in Linear RNNs Through Negative Eigenvalues. In *The Thirteenth International Conference on Learning Representations (ICLR'25)*. ICLR, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Learning and Modeling (COLM 2024)*, 2024.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022.

Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*, 2020.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, 2024.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36:24892–24928, 2023.

Shanda Li, Chong You, Guru Guruganesh, Joshua Ainslie, Santiago Ontañón, Manzil Zaheer, Sumit Sanghai, Yiming Yang, Sanjiv Kumar, and Srinadh Bhojanapalli. Functional interpolation for relative positions improves long context transformers. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=rR03qFesqk.

Zhixuan Lin, Evgenii Nikishin, Xu Owen He, and Aaron Courville. Forgetting transformer: Softmax attention with a forget gate. *arXiv preprint arXiv:2503.02130*, 2025.

B. Liu, J. Ash, S. Goel, A. Krishnamurthy, and C. Zhang. Transformers Learn Shortcuts to Automata. In *The Eleventh International Conference on Learning Representations (ICLR'23)*. ICLR, 2023.

W. Merrill, J. Petty, and A. Sabharwal. The Illusion of State in State-Space Models. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*, volume 251 of *Proceedings of Machine Learning Research*. PMLR, 2024.

Nicola Muca Cirone, Antonio Orvieto, Benjamin Walker, Cristopher Salvi, and Terry Lyons. Theoretical foundations of deep selective state-space models. *Advances in Neural Information Processing Systems*, 37:127226–127272, 2024.

D. Okpekpe and A. Orvieto. When recalling in-context, Transformers are not SSMs. *arXiv:2508.19029 [cs.LG]*, 2025.

Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

A. Orvieto and R. Gower. In search of adam's secret sauce. *arXiv:2505.21829 [cs.LG]*, 2025.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pp. 26670–26698. PMLR, 2023.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. *arXiv preprint arXiv:2103.02143*, 2021.

M. Poli, A. W. Thomas, E. Nguyen, P. Ponnusamy, B. Bjö"rn Deiseroth, K. Kersting, T. Suzuki, B. Hie, S. Ermon, C. Re, C. Zhang, and S. Massaroli. Mechanistic design and scaling of hybrid architectures. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*, volume 251 of *Proceedings of Machine Learning Research*. PMLR, 2024.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

J. Siems, T. Carstensen, A. Zela, F. Hutter, M. Pontil, and R. Grazzi. DeltaProduct: Increasing the expressivity of deltanet through products of householders. *arXiv:2502.10297 [cs.LG]*, 2025a.

Julien Siems, Timur Carstensen, Arber Zela, Frank Hutter, Massimiliano Pontil, and Riccardo Grazzi. Deltaproduct: Improving state-tracking in linear rnns via householder products. *arXiv preprint arXiv:2502.10297*, 2025b.

Julius O Smith III. Spectral audio signal processing. *(No Title)*, 2011.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

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, \Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

S. Yang and Y. Zhang. Fla: A triton-based library for hardware-efficient implementations of linear attention mechanism, January 2024. URL https://github.com/fla-org/flash-linear-attention.

S. Yang, B. Wang, Y. Shen, R. Panda, and Y. Kim. Gated Linear Attention Transformers with Hardware-Efficient Training. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*, volume 251 of *Proceedings of Machine Learning Research*. PMLR, 2024a.

S. Yang, Y. Shen, K. Wen, S. Tan, M. Mishra, L. Ren, R. Panda, and Y. Kim. PaTH Attention: Position encoding via accumulating householder transformations. *arXiv:2505.16381 [cs.CL]*, 2025a.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule, 2024. *URL https://arxiv. org/abs/2412.06464*.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *arXiv preprint arXiv:2406.06484*, 2024b.

Songlin Yang, Yikang Shen, Kaiyue Wen, Shawn Tan, Mayank Mishra, Liliang Ren, Rameswar Panda, and Yoon Kim. Path attention: Position encoding via accumulating householder transformations. *arXiv preprint arXiv:2505.16381*, 2025b.

A Mathematical Derivations and Proofs

A.1 *RoPE* as Imaginary-valued Linear Transformer

We start by unrolling the linear transformers recurrence:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1}\bar{\boldsymbol{R}} + \boldsymbol{v}_t\tilde{\boldsymbol{k}}_t^\top, \quad \boldsymbol{o}_t = \mathcal{R}\{\boldsymbol{S}_t\tilde{\boldsymbol{q}}_t\}$$

$$\boldsymbol{o}_t = \mathcal{R}\left\{\sum_{\tau=1}^{t}\boldsymbol{v}_\tau\tilde{\boldsymbol{k}}_\tau^\top\bar{\boldsymbol{R}}^{t-\tau}\tilde{\boldsymbol{q}}_t\right\} = \sum_{\tau=1}^{t}\boldsymbol{v}_\tau\mathcal{R}\left\{\tilde{\boldsymbol{k}}_\tau^\top\bar{\boldsymbol{R}}^{t-\tau}\tilde{\boldsymbol{q}}_t\right\}$$

Therefor, the attention score applied to value $\boldsymbol{v}_\tau$ is:

$$\mathbf{Att}_{t\tau} = \mathcal{R}\left\{\tilde{\boldsymbol{k}}_\tau^\top\bar{\boldsymbol{R}}^{t-\tau}\tilde{\boldsymbol{q}}_t\right\}$$

where $\mathcal{R}$ extracts the real part of attention. Since $\bar{\mathbf{R}}$ is diagonal, we can expand the expression as:

$$\begin{aligned}
\mathbf{Att}_{t\tau} &= \mathcal{R}\left\{\sum_{n=1}^{d/2}(\tilde{\boldsymbol{q}}_{t,n}^R + i\,\tilde{\boldsymbol{q}}_{t,n}^I)\cdot e^{i\omega_n(t-\tau)}\cdot(\tilde{\boldsymbol{k}}_{\tau,n}^R + i\,\tilde{\boldsymbol{k}}_{\tau,n}^I)\right\} \\
&= \mathcal{R}\left\{\sum_{n=1}^{d/2}|\tilde{\boldsymbol{q}}_{t,n}|\,e^{-i\angle\tilde{\boldsymbol{q}}_{t,n}}\cdot e^{i\omega_n(t-\tau)}\cdot|\tilde{\boldsymbol{k}}_{\tau,n}|\,e^{-i\angle\tilde{\boldsymbol{k}}_{\tau,n}}\right\} \\
\mathbf{Att}_{t\tau} &= \mathcal{R}\left\{\sum_{n=1}^{d/2}|\tilde{\boldsymbol{q}}_{t,n}|\,|\tilde{\boldsymbol{k}}_{\tau,n}|\,e^{i\left(\omega_n(t-\tau)-\angle\tilde{\boldsymbol{q}}_{t,n}-\angle\tilde{\boldsymbol{k}}_{\tau,n}\right)}\right\} \\
\mathbf{Att}_{t\tau} &= \sum_{n=1}^{d/2}|\tilde{\boldsymbol{q}}_{t,n}|\,|\tilde{\boldsymbol{k}}_{\tau,n}|\,\cos\left(\omega_n(t-\tau)-\angle\tilde{\boldsymbol{q}}_{t,n}-\angle\tilde{\boldsymbol{k}}_{\tau,n}\right) \qquad (20)
\end{aligned}$$

where $\angle\tilde{\boldsymbol{q}}_{t,n}$ and $\angle\tilde{\boldsymbol{k}}_{\tau,n}$ denote the complex phases (angles) of the n-th component of $\tilde{\boldsymbol{q}}_t$ and $\tilde{\boldsymbol{k}}_\tau$, respectively. Equation 20 shows that an imaginary forget gate rotates the query/key pairs at each index n with a distinct frequency ω_n. We now demonstrate that this is equivalent to applying *RoPE*. Replacing the cosine in Equation 20 with its matrix multiplication equivalent:

$$\cos\left(\omega_n(t-\tau)-\angle\tilde{\boldsymbol{q}}_{t,n}-\angle\tilde{\boldsymbol{k}}_{\tau,n}\right) = \begin{bmatrix}\cos(\angle\tilde{\boldsymbol{q}}_{t,n})\\ \sin(\angle\tilde{\boldsymbol{q}}_{t,n})\end{bmatrix}^\top\begin{bmatrix}\cos(\omega_n(t-\tau)) & -\sin(\omega_n(t-\tau))\\ \sin(\omega_n(t-\tau)) & \cos(\omega_n(t-\tau))\end{bmatrix}\begin{bmatrix}\cos(\angle\tilde{\boldsymbol{k}}_{\tau,n})\\ \sin(\angle\tilde{\boldsymbol{k}}_{\tau,n})\end{bmatrix}$$

Plugging above in Equation 20 we achieve:

$$\begin{aligned}
\mathbf{Att}_{t\tau} &= \sum_{n=1}^{d/2}|\tilde{\boldsymbol{q}}_{t,n}|\,|\tilde{\boldsymbol{k}}_{\tau,n}|\begin{bmatrix}\cos(\angle\tilde{\boldsymbol{q}}_{t,n})\\ \sin(\angle\tilde{\boldsymbol{q}}_{t,n})\end{bmatrix}^\top\begin{bmatrix}\cos(\omega_n(t-\tau)) & -\sin(\omega_n(t-\tau))\\ \sin(\omega_n(t-\tau)) & \cos(\omega_n(t-\tau))\end{bmatrix}\begin{bmatrix}\cos(\angle\tilde{\boldsymbol{k}}_{\tau,n})\\ \sin(\angle\tilde{\boldsymbol{k}}_{\tau,n})\end{bmatrix} \\
\mathbf{Att}_{t\tau} &= \sum_{n=1}^{d/2}|\tilde{\boldsymbol{q}}_{t,n}|\begin{bmatrix}\cos(\angle\tilde{\boldsymbol{q}}_{t,n})\\ \sin(\angle\tilde{\boldsymbol{q}}_{t,n})\end{bmatrix}^\top\begin{bmatrix}\cos(\omega_n(t-\tau)) & -\sin(\omega_n(t-\tau))\\ \sin(\omega_n(t-\tau)) & \cos(\omega_n(t-\tau))\end{bmatrix}|\tilde{\boldsymbol{k}}_{\tau,n}|\begin{bmatrix}\cos(\angle\tilde{\boldsymbol{k}}_{\tau,n})\\ \sin(\angle\tilde{\boldsymbol{k}}_{\tau,n})\end{bmatrix} \\
\mathbf{Att}_{t\tau} &= \sum_{n=1}^{d/2}\begin{bmatrix}\tilde{\boldsymbol{q}}_{t,n}^R\\ \tilde{\boldsymbol{q}}_{t,n}^I\end{bmatrix}^\top\begin{bmatrix}\cos(\omega_n(t-\tau)) & -\sin(\omega_n(t-\tau))\\ \sin(\omega_n(t-\tau)) & \cos(\omega_n(t-\tau))\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{k}}_{t,n}^R\\ \tilde{\boldsymbol{k}}_{t,n}^I\end{bmatrix} \qquad (21)
\end{aligned}$$

Using the definition of:

$$\boldsymbol{q}_t = \bigoplus_{n=1}^{d/2}\begin{bmatrix}\tilde{\boldsymbol{q}}_{t,n}^R\\ \tilde{\boldsymbol{q}}_{t,n}^I\end{bmatrix}, \quad \boldsymbol{k}_t = \bigoplus_{n=1}^{d/2}\begin{bmatrix}\tilde{\boldsymbol{k}}_{t,n}^R\\ \tilde{\boldsymbol{k}}_{t,n}^I\end{bmatrix}.$$

we can write equation 21 as:

$$\mathbf{Att}_{t\tau} = \sum_{n=1}^{d/2}\boldsymbol{q}_{t,n}\boldsymbol{R}_{\omega_n}^{t-\tau}\boldsymbol{k}_{\tau,n}$$

which is theoretically equivalent to applying *RoPE* to query/key pairs $\boldsymbol{q}_t, \boldsymbol{k}_\tau$. *RoPE* interleaves the real and imaginary parts of complex queries and keys across the hidden dimension, then applies 2D rotations to each pair.

A.2 SOFTMAX ATTENTION DERIVATION

We start with the definition of softmax attention:

$$\boldsymbol{o}_t = \frac{\boldsymbol{s}_t}{\boldsymbol{z}_t}, \quad \boldsymbol{s}_t = \sum_{\tau=1}^{t} \exp\Big(\tfrac{1}{\sqrt{d}} \boldsymbol{q}_t^\top \boldsymbol{k}_\tau\Big) \cdot \boldsymbol{v}_\tau, \quad \boldsymbol{z}_t = \sum_{\tau=1}^{t} \exp\Big(\tfrac{1}{\sqrt{d}} \boldsymbol{q}_t^\top \boldsymbol{k}_\tau\Big),$$

where $\boldsymbol{q}_t, \boldsymbol{k}_\tau \in \mathbb{R}^d$. For simplicity, we omit the normalization factor $1/\sqrt{d}$ and first focus on the numerator of the output, specifically the exponential kernel. As in equation 2, the denominator scaling can be handled separately through an external state $\boldsymbol{z}_t$.

To approximate the exponential kernel $\exp(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau)$, we use Random Fourier Features (RFF) (Rahimi & Recht, 2007) with frequencies $\boldsymbol{\omega} \in \mathbb{R}^d \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. The feature map is defined as

$$\phi_{\boldsymbol{\omega}}(\boldsymbol{x}) = \exp\left(\frac{\|\boldsymbol{x}\|_2^2}{2} + i\boldsymbol{\omega}^\top \boldsymbol{x}\right),$$

so that

$$\exp(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau) = \mathcal{R}\big\{\mathbb{E}_{\boldsymbol{\omega}\sim\mathcal{N}(0,\sigma^2\boldsymbol{I})}\big[\phi_{\boldsymbol{\omega}}(\boldsymbol{q}_t)^\top \phi_{\boldsymbol{\omega}}(\boldsymbol{k}_\tau)\big]\big\},$$

for $\sigma = 1$. By applying this feature map, the linear attention formulation in equation 2, we can approximate the exponential kernel in softmax attention. Continuing the approximation:

$$\exp\left(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\right) = \exp\left(\frac{\|\boldsymbol{q}_t\|_2^2 + \|\boldsymbol{k}_\tau\|_2^2}{2}\right) \cdot \mathcal{R}\big\{\mathbb{E}_{\boldsymbol{\omega}\sim\mathcal{N}(0,\boldsymbol{I})}\big[\exp(i\boldsymbol{\omega}^\top \boldsymbol{q}_t)\exp(-i\boldsymbol{\omega}^\top \boldsymbol{k}_\tau)\big]\big\}.$$

Let $\boldsymbol{\omega}_j \sim \mathcal{N}\left(0, \sigma^2 \boldsymbol{I}\right)$ for $j \in \{1, 2, \ldots, D\}$. Then due to the law of large numbers we have:

$$\exp\left(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\right) = \exp\left(\frac{\|\boldsymbol{q}_t\|_2^2 + \|\boldsymbol{k}_\tau\|_2^2}{2}\right) \cdot \mathcal{R}\left\{\lim_{D\to\infty} \frac{1}{D} \sum_{j=1}^{D} \exp\left(i\boldsymbol{\omega}_j^\top \boldsymbol{q}_t\right) \cdot \exp\left(-i\boldsymbol{\omega}_j^\top \boldsymbol{k}_\tau\right)\right\}.$$

Therefore, we can approximate $\exp\left(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\right)$ as the dot product of the random exponential projection of the query and the key using D random $\boldsymbol{\omega}_j$s:

$$\hat{\boldsymbol{s}}_t^D = \frac{1}{D} \sum_{\tau=1}^{t} \sum_{j=1}^{D} \exp\left(\frac{\|\boldsymbol{q}_t\|_2^2 + \|\boldsymbol{k}_\tau\|_2^2}{2}\right) \exp\left(i\boldsymbol{\omega}_j^\top \boldsymbol{q}_t\right) \exp\left(-i\boldsymbol{\omega}_j^\top \boldsymbol{k}_\tau\right) \cdot \boldsymbol{v}_\tau.$$

This allows us to compute the softmax attention as the linear attention parameterized by:

$$\phi(\boldsymbol{q}_t) = \exp\left(\frac{\|\boldsymbol{q}_t\|_2^2}{2}\right) \cdot \exp(i\boldsymbol{\Omega}^\top \boldsymbol{q}_t), \quad \phi(\boldsymbol{k}_\tau) = \exp\left(\frac{\|\boldsymbol{k}_t\|_2^2}{2}\right) \cdot \exp(-i\boldsymbol{\Omega}^\top \boldsymbol{k}_\tau),$$

with $\lim_{D\to\infty} \mathcal{R}\left\{\hat{\boldsymbol{s}}_t^D\right\} = \sum_{\tau=1}^{t} \exp\big(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\big) \cdot \boldsymbol{v}_\tau$ and $\boldsymbol{\Omega} = [\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_D]$. Omitting the superscript D for simplifying the notation, let us focus on one random feature $\boldsymbol{\omega}_j$ and its contribution to the output:

$$\hat{\boldsymbol{s}}_{t,j} = \sum_{\tau=1}^{t} \exp\left(\frac{\|\boldsymbol{q}_t\|_2^2}{2}\right) \exp\left(\frac{\|\boldsymbol{k}_\tau\|_2^2}{2}\right) \exp\left(i\boldsymbol{\omega}_j^\top \boldsymbol{q}_t\right) \exp\left(-i\boldsymbol{\omega}_j^\top \boldsymbol{k}_\tau\right) \cdot \boldsymbol{v}_\tau.$$

In this case, we have $\hat{\boldsymbol{s}}_t^D = \frac{1}{D}\hat{\boldsymbol{S}}_t^D \mathbf{1}$, where $\hat{\boldsymbol{S}}_t^D = [\hat{\boldsymbol{s}}_{t,1} \quad \hat{\boldsymbol{s}}_{t,2} \quad \ldots \quad \hat{\boldsymbol{s}}_{t,D}] \in \mathbb{C}^{d\times D}$. Now note that we have:

$$\hat{\boldsymbol{s}}_{t,j} = \sum_{\tau=1}^{t-1} \exp\Big(\tfrac{\|\boldsymbol{q}_t\|_2^2 - \|\boldsymbol{q}_{t-1}\|_2^2}{2}\Big) \exp\big(i\boldsymbol{\omega}_j^\top \boldsymbol{q}_{t-1}\big) \exp\big(i\boldsymbol{\omega}_j^\top (\boldsymbol{q}_t - \boldsymbol{q}_{t-1})\big) \exp\big(-i\boldsymbol{\omega}_j^\top \boldsymbol{k}_\tau\big) \cdot \boldsymbol{v}_\tau \quad (22)$$

$$+ \exp\Big(\tfrac{\|\boldsymbol{q}_t\|_2^2}{2}\Big) \exp\Big(\tfrac{\|\boldsymbol{k}_t\|_2^2}{2}\Big) \exp\big(i\boldsymbol{\omega}_j^\top (\boldsymbol{q}_t - \boldsymbol{k}_t)\big) \cdot \boldsymbol{v}_t. \quad (23)$$

$$= \exp\Big(\tfrac{\|\boldsymbol{q}_t\|_2^2 - \|\boldsymbol{q}_{t-1}\|_2^2}{2}\Big) \exp\big(i\boldsymbol{\omega}_j^\top (\boldsymbol{q}_t - \boldsymbol{q}_{t-1})\big)\, \hat{\boldsymbol{s}}_{t-1}^j + \phi_{\boldsymbol{\omega}_j}(\boldsymbol{q}_t) \cdot \phi_{\boldsymbol{\omega}_j}(\boldsymbol{k}_t) \cdot \boldsymbol{v}_t \quad (24)$$

Note that the real exponential component in equation 24 can introduce instability to the recurrence. Therefore, following the standard in both linear transformers (Yang et al., 2024b; 2023; Yang et al.;

Lin et al., 2025) and deep softmax transformers (Henry et al., 2020), we assume L_2 normalization over the query and the key, i.e., $\|\boldsymbol{q}_t\|_2 = \|\boldsymbol{q}_{t-1}\|_2$. Thus, recurrence presented in equation 24 simplifies to:

$$\hat{\boldsymbol{s}}_{t,j} = \exp\left(i\boldsymbol{\omega}_j^\top (\boldsymbol{q}_t - \boldsymbol{q}_{t-1})\right) \hat{\boldsymbol{s}}_{t-1,j} + \phi_{\boldsymbol{\omega}_j}(\boldsymbol{q}_t) \cdot \phi_{\boldsymbol{\omega}_j}(\boldsymbol{k}_t) \cdot \boldsymbol{v}_t, \tag{25}$$

with $\hat{\boldsymbol{s}}_{t,j}$ being the j^{th} column of $\hat{\boldsymbol{S}}_t^D$ is scaled by the values $\exp\left(i\boldsymbol{\omega}_j^\top (\boldsymbol{q}_t - \boldsymbol{q}_{t-1})\right)$. Therefore, we can write the recurrence over $\hat{\boldsymbol{S}}_t$ as:

$$\hat{\boldsymbol{S}}_t^D = \hat{\boldsymbol{S}}_{t-1}\bar{\boldsymbol{R}}_t + \boldsymbol{v}_t \left(\phi(\boldsymbol{q}_t) \circ \phi(\boldsymbol{k}_t)\right)^\top, \quad \hat{\boldsymbol{s}}_t^D = \frac{1}{D}\hat{\boldsymbol{S}}_t^D \mathbf{1}.$$

where $\phi(x)$ is a vector with its j^{th} element equal to $\phi_{\boldsymbol{\omega}_j}(x)$, and $\bar{\boldsymbol{R}}_t$ is:

$$\bar{\boldsymbol{R}}_t = \exp(i\boldsymbol{\Omega}^\top (\boldsymbol{q}_t - \boldsymbol{q}_{t-1})) \tag{26}$$

Focusing on equation 26, we observe that exponential kernel in softmax attention implicitly applies a form of input-dependent *(Selective) RoPE* (see Sec. 2.2). However, instead of learning the frequencies $\boldsymbol{\Omega}$, they are randomly sampled from a normal distribution.

Similarly, we can also approximate the normalizing factor $\boldsymbol{z}_t$ as:

$$\hat{\boldsymbol{z}}_t^D = \frac{1}{D}\sum_{\tau=1}^{t}\sum_{j=1}^{D} \exp\left(\frac{\|\boldsymbol{q}_t\|_2^2 + \|\boldsymbol{k}_\tau\|_2^2}{2}\right) \exp\left(i\boldsymbol{\omega}_j^\top \boldsymbol{q}_t\right) \exp\left(-i\boldsymbol{\omega}_j^\top \boldsymbol{k}_\tau\right).$$

Separating the contribution of each random feature, we have:

$$\hat{\boldsymbol{z}}_{t,j} = \sum_{\tau=1}^{t} \exp\left(\frac{\|\boldsymbol{q}_t\|_2^2}{2}\right) \exp\left(\frac{\|\boldsymbol{k}_\tau\|_2^2}{2}\right) \exp\left(i\boldsymbol{\omega}_j^\top \boldsymbol{q}_t\right) \exp\left(-i\boldsymbol{\omega}_j^\top \boldsymbol{k}_\tau\right).$$

Finally, defining $\hat{\boldsymbol{Z}}_t^D = [\hat{\boldsymbol{z}}_{t,1} \quad \hat{\boldsymbol{z}}_{t,2} \quad \dots \quad \hat{\boldsymbol{z}}_{t,D},]$ we arrive at a similar result. The full recurrence of softmax attention, therefore, can be written as:

$$\hat{\boldsymbol{S}}_t^D = \hat{\boldsymbol{S}}_{t-1}^D\bar{\boldsymbol{R}}_t + \boldsymbol{v}_t \left(\phi(\boldsymbol{q}_t) \circ \phi(\boldsymbol{k}_t)\right)^\top, \quad \hat{\boldsymbol{Z}}_t^D = \hat{\boldsymbol{Z}}_{t-1}^D\bar{\boldsymbol{R}}_t + \phi(\boldsymbol{q}_t) \circ \phi(\boldsymbol{k}_t), \quad \hat{\boldsymbol{o}}_t = \frac{\hat{\boldsymbol{S}}_t^D\mathbf{1}}{\hat{\boldsymbol{z}}_t^D\mathbf{1}}.$$

which again highlights the importance of the gate $\bar{\boldsymbol{R}}$ as selective rotation.

A.3 Optimal variance for RFF: Proof of Theorem 1

Proof 1 *We start by writing down the error:*

$$\begin{aligned}
ERR\left[\boldsymbol{q}_t, \boldsymbol{k}_\tau\right] =& \frac{e^2}{D^2}\sum_{j,j'=1} \mathbb{E}\left[\mathcal{R}\left[\exp\left(i\left(\omega_j + \omega_{j'}\right)^\top (\boldsymbol{q}_t - \boldsymbol{k}_\tau)\right)\right]\right] \\
&- \frac{2e}{D}\sum_{j=1} \mathbb{E}\left[\mathcal{R}\left[\exp\left(i\omega_j^\top (\boldsymbol{q}_t - \boldsymbol{k}_\tau)\right)\right]\right] \exp\left(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\right) + const. \\
=& \frac{e^2}{D}\mathbb{E}\left[\cos^2\left(i\omega^\top (\boldsymbol{q}_t - \boldsymbol{k}_\tau)\right)\right] + \frac{e^2\left(D^2 - D\right)}{D^2}\mathbb{E}\left[\cos\left(i\omega^\top (\boldsymbol{q}_t - \boldsymbol{k}_\tau)\right)\right]^2 \\
&- 2e \cdot \mathbb{E}\left[\cos\left(i\omega^\top (\boldsymbol{q}_t - \boldsymbol{k}_\tau)\right)\right] \exp\left(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau\right) + const.,
\end{aligned}$$

where the const. term corresponds to the terms constant w.r.t. the variance of the distribution σ^2. Plugging in the expectation of the $\cos(.)$ and $\cos^2(.)$ functions (Choromanski et al., 2021), we get the following optimization problem:

$$\min_\sigma \left[\frac{e^{2-4\sigma^2} \cdot \exp\left(-4\sigma^2\xi\right)}{2D} + \frac{D-1}{D}e^{2-2\sigma^2}\exp\left(-2\sigma^2\xi\right) - 2e^{1-\sigma^2}\exp\left(\left(1-\sigma^2\right)\xi\right)\right],$$

where for simplicity, we set $\boldsymbol{q}_t^\top \boldsymbol{k}_\tau = \xi \in [0, 1]$. Since in most cases, D is a sizable number, we try to solve this optimization problem in the limit $D \to \infty$, which is equivalent to:

$$\min_\sigma \left[e^{2-2\sigma^2(1+\xi)} - 2e^{\left(1-\sigma^2\right)(1+\xi)}\right],$$

with the optimal value equal to:

$$\sigma = \sqrt{\frac{1-\xi}{1+\xi}}.$$

Considering normalized queries and keys $||\boldsymbol{k}_t|| = ||\boldsymbol{q}_t|| = 1$ *we can replace the* $\xi = \boldsymbol{q}_t^\top \boldsymbol{k}_\tau$ *with* $\cos(\theta)$ *therefor above also simplifies to:*

$$\sigma = \sqrt{\frac{1-\cos(\theta)}{1+\cos(\theta)}} = \tan(\theta/2).$$

This completes our proof. ■

A visualization of the temperature distribution in *Selective RoPE* compared to standard *RoPE* is shown in fig. 7.

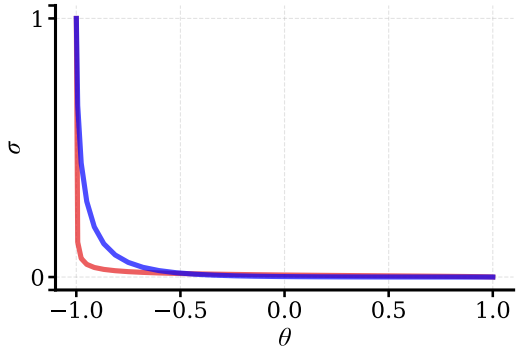

Figure 7: Comparison of temperature distributions in *RoPE* and *Selective RoPE*.

A.4 Complex rotations and Householder matrices

Another approach towards introducing rotations to the quries and keys is using Householder reflection matrices Yang et al. (2024b; 2025b). In this approach, the rotation of the query and key pair is limited to a single reflection along the direction of an input-dependent vector. Specifically, let $\boldsymbol{w}_t$ be an input-dependent unit vector. Then, the positional information is encoded through the product of Householder reflection matrices as:

$$\boldsymbol{q}_t^\top \boldsymbol{R}_{t:\tau} \boldsymbol{k}_\tau = \boldsymbol{q}_t^\top \left(\prod_{\kappa=\tau+1}^{t} \left(\boldsymbol{I} - 2\beta_\kappa \cdot \boldsymbol{w}_\kappa \boldsymbol{w}_\kappa^\top \right) \right) \boldsymbol{k}_\tau.$$

Therefore, the positional information between the t^{th} and τ^{th} token is encoded through a rotation consisting of $t - \tau$ reflections.

Conveniently, we can also write the complex diagonal rotation matrix in *Selective RoPE* in terms of the product of Householder matrices. Specifically, we can write the realification of the rotation matrix in equation 5 as the product of d Householer reflections, each of which performs the reflection over a single pair of adjacent elements:

$$\boldsymbol{R}_t = \prod_{j=1}^{d} \left(\boldsymbol{I} - 2 \cdot \begin{bmatrix} \boldsymbol{0}_j \\ 1 \\ 0 \\ \boldsymbol{0}_{d-j-2} \end{bmatrix} \begin{bmatrix} \boldsymbol{0}_j \\ 1 \\ 0 \\ \boldsymbol{0}_{d-j-2} \end{bmatrix}^\top \right) \left(\boldsymbol{I} - 2 \begin{bmatrix} \boldsymbol{0}_j \\ \cos(\omega_{t,j}/2) \\ \sin(\omega_{t,j}/2) \\ \boldsymbol{0}_{d-j-2} \end{bmatrix} \begin{bmatrix} \boldsymbol{0}_j \\ \cos(\omega_{t,j}/2) \\ \sin(\omega_{t,j}/2) \\ \boldsymbol{0}_{d-j-2} \end{bmatrix}^\top \right),$$

where we define $\boldsymbol{0}_m \in \mathbb{R}^m$ as a vector with all zeros. Assuming we split adjacent elements in the query/key into the real and imaginary components, then *Selective RoPE* is performing two reflections over each adjacent element pair of the input, with one of them a parametric reflection, and the other negating the first element.

This interpretation also explains why we gain more expressivity when using *Selective RoPE*: due to the block-diagonal structure, there is a channel mixing happening between the adjacent query/key elements. Channel mixing is a key component in improving the expressivity of sequence models (Muca Cirone et al., 2024), thus improving the state-tracking abilities of the network (Siems et al., 2025b).

B Background work on Diagonal Structured SSMs

Structured SSMs are motivated based on the continuous-time SSM representations and can be presented as linear ODE or in parallel convolution form (Gu et al., 2022):

$$\frac{d\boldsymbol{s}(t)}{dt} = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{B}\boldsymbol{x}(t), \quad \boldsymbol{o}(t) = \boldsymbol{C}^\top \boldsymbol{s}(t), \quad K(t) = \boldsymbol{C}^\top e^{\boldsymbol{A}t}\boldsymbol{B}, \quad \boldsymbol{o}(t) = K * \boldsymbol{x} \tag{27}$$

Here, the parameters are defined as $\boldsymbol{A} \in \mathbb{C}^{N\times N}, \boldsymbol{B}, \boldsymbol{C} \in \mathbb{C}^N$. Above can be mapped into the discrete domain using different discretizations, resulting in:

$$\boldsymbol{s}_t = \bar{\boldsymbol{A}}\boldsymbol{s}_{t-1} + \bar{\boldsymbol{B}}\boldsymbol{x}_t, \quad \boldsymbol{o}_t = \bar{\boldsymbol{C}}^\top \boldsymbol{s}_t \tag{28}$$

where discrete state values $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \bar{\boldsymbol{C}}$ are dependent on their continuous forms $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ via choice of discretization and scalar sampling step Δ. Diagonal SSMs, such as S4D (Gu et al., 2022), use a diagonal state matrix $\boldsymbol{A}$ with Zero-Order Hold (ZOH) discretization (Kalman, 1960), resulting in:

$$\boldsymbol{s}_t = e^{\boldsymbol{A}\Delta}\boldsymbol{s}_{t-1} + \boldsymbol{B}\boldsymbol{x}_t, \quad \boldsymbol{o}_t = \boldsymbol{C}^\top(e^{\boldsymbol{A}\Delta} - \boldsymbol{I})\boldsymbol{A}^{-1}\boldsymbol{s}_t, \quad \boldsymbol{A} = -\exp(\boldsymbol{A}_R) + i\boldsymbol{A}_I \tag{29}$$

where $\boldsymbol{A}^R$ and $\boldsymbol{A}^I$ are the real and imaginary parts of the diagonal state matrix $\boldsymbol{A}$, respectively. For stability, S4D enforces negative real parts via $-\exp(\boldsymbol{A}^R)$, with $\boldsymbol{A}^R, \boldsymbol{A}^I$ initialized using HIPPO (Gu et al., 2020) or close to unit circle initialization (Orvieto et al., 2023).

B.1 Relationship between *Selective RoPE* and FoX

FoX (Lin et al., 2025) is a softmax transformer that augments attention with a real-valued forget gate inspired by GLA. Its attention can be written as:

$$\boldsymbol{q}_t, \boldsymbol{k}_t, \boldsymbol{v}_t = \boldsymbol{W}_q\boldsymbol{x}_t, \boldsymbol{W}_k\boldsymbol{x}_t, \boldsymbol{W}_v\boldsymbol{x}_t, \qquad \boldsymbol{o}_t = \frac{\sum_{\tau=1}^t \exp(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau + \prod_{\kappa=\tau}^t a_\kappa)\boldsymbol{v}_\tau}{\sum_{\tau=1}^t \exp(\boldsymbol{q}_t^\top \boldsymbol{k}_\tau + \prod_{\kappa=\tau}^t a_\kappa)}. \tag{30}$$

Here, the gate decays the norm of querykey pairs through a selective decay parameterized in log-space, $a_t = \log(f_t)$. This enhances the forgetting capability of transformers, addressing our earlier observation in section 3.1 that softmax alone preserves norms and thus cannot forget. Interestingly, in the softmax setting, *Selective RoPE* closely parallels FoX: it can be seen as replacing the decay term a_t with a rotation matrix $\boldsymbol{R}_t$.

C Neural Architecture of GLA with *Selective RoPE*

D Implementation

We provide a PyTorch implementation of Selective RoPE in Figure 9.

E Experimental Details

In this section we provide additional details on our experimental setup for the tasks considered in the paper.

E.1 Language Modeling

E.1.1 Setup and hyperparameters.

We use PlainLM (Ajroldi, 2024) together with an adapted version of `flash-linear-attention` for all of our language model trainings. We train on $>$ 80GB

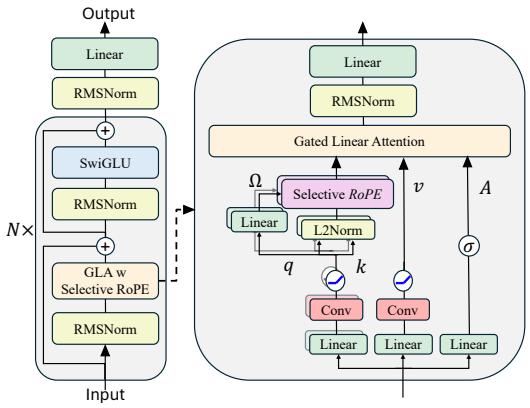

Figure 8: neural Architecture of GLA+*Selective RoPE*

VRAM GPUs incuding NVIDIA A100, H100 and B200. One model training (370M parameters, 35B tokens) is performed on a single node with 4 to 8 of such GPUs and takes anywhere from 48 hours (on 4 A100) to 9 hours on 8 B200. We use Distributed Data Parallel (DDP) for multi-GPU training.

Table 4: Optimizer and learning-rate schedule hyperparameters for language modeling.

		Optimizer
Parameter	Symbol	Value
Base learning rate (candidates)	η	`[5e-4, 1e-3, 2e-3, 4e-3, 8e-3, 1.6e-2]`
Adam β_1	β_1	0.9
Adam β_2	β_2	0.95
Weight decay	λ	0.1
Numerical epsilon	ϵ	1×10^{-8}
Gradient clipping (global norm)	clip_{ℓ_2}	1.0
		LR Schedule / Training Horizon
LR start (schedule)	η_{start}	1e-5
LR end (schedule)	η_{end}	1e-4
Warmup (fraction of steps)	–	0.1
Total optimizer steps	T	66,758

E.2 Synthetic Tasks

E.2.1 MAD

For MAD, we take the implementation from `mad_lab` and implement *Selective RoPE* in GLA. We follow the exact experimental setup outlined in the paper (Poli et al., 2024) and run all variations of task difficulty and optimizer hyperparameters which results in 66 task settings $\times$ 6 optimizer settings $=$ 396 trained models per considered setting (i.e., GLA with *Selective RoPE*, RoPE or NoPE). We provide the logs from the experiments in our supplementary.

E.2.2 State Tracking

For state tracking we adopt the exact experimental setup as described in DeltaProduct (Siems et al., 2025a) and Grazzi et al. (2025).

E.2.3 MQAR

We have carefully followed the training recipe of Arora et al. (2024a) for all models including: GLA (Yang et al., 2024a), DeltaNet (Yang et al., 2024b), Mamba2 (Dao & Gu, 2024) and Transformer++ (Touvron et al., 2023). The learning rate for all models were swiped within the range of $[0.0001, 0.01]$ for 8 different values per each model. All other configuration and the model dimensions were remained the same as original reference Arora et al. (2024a).

Table 5: Training state tracking configuration.

Training Loop	
Parameter	Value
Epochs	100
Batch size	4096
Optimization	
Learning rate	1e-3
β_1	0.9
β_2	0.999
Optimizer ϵ	1e-8
Weight decay	1e-6
LR scheduler	cosine
Precision / Compile	
Mixed precision	true
DType	bfloat16
Data	
Train set size	2,000,000 sequences
Train sequence length	128 tokens
Eval set size	500,000 sequences
Eval sequence length	512 tokens
Seeds & Eval	
Seeds	`[555, 666, 777, 888, 999]`
Eval batch size	128

E.2.4 COPYING

Table 6: Optimizer and Data parameters for Copying

Optimizer	
Learning rate	5.0e-5
Weight decay	0.1
β_1	0.9
β_2	0.999
Optimizer ϵ	1.0e-8
Gradient clipping (global norm)	1.0
Scheduler	
Scheduler	linear
Warmup (fraction of steps)	0.1
Seeds & Eval	
Seed	42
Eval batch size	256
Data	
Vocab size	26
n-gram	0
Answer length	0
Train task	`copy`
Eval task	`copy`
Sequence length	420
Min length (train)	2
Max length (train)	64
Min length (eval)	2
Max length (eval)	512
Sampler type	sequential
Sampler seed	null

E.3 THE USE OF LARGE LANGUAGE MODELS (LLMS)

While preparing this manuscript, we limitedly used Large Language Models (LLMs). Their role was restricted to assisting with editing and polishing the writing, such as improving clarity, grammar, and flow. All conceptual ideas, methods, experiments, and analyses presented in this paper are entirely the work of the authors. No ideas, algorithms, or research contributions were generated by an LLM.

The LLM served only as a tool to refine the presentation of the text without influencing the substance of the research.

```python
from fla.modules.convolution import ShortConvolution
from einops import rearrange
import torch
import torch.nn as nn
from .chunked_linear import ChunkedLinear

class SelectiveRoPE(nn.Module):
    def __init__(
        self,
        head_dim: int,
        num_heads: int = 1,
        dtype: torch.dtype | None = None,
        d_conv: int = 4,
        temp_type: str = "rope",
        temp_theta: float = 500000,
        temp_max: float = 1.0,
        temp_grad: bool = False,
        is_softmax: bool = False,
        phi_conv_activation: str | None = None,
    ):
        super().__init__()
        self.head_dim = head_dim
        self.num_heads = num_heads
        self.is_softmax = is_softmax
        pe_dim = head_dim
        self.phi_proj = ChunkedLinear(2 * pe_dim, pe_dim,
            num_heads=num_heads, bias=False, random_init=True,
            rank=-1,
        )
        self.phi_conv1d = ShortConvolution(
            hidden_size=num_heads * pe_dim,
            kernel_size=d_conv, bias=False,
            activation=phi_conv_activation, dtype=dtype,
        )
        self.temperature = nn.Parameter(
            rotary_temperature(temp_type, temp_theta, head_dim, temp_max).reshape(1, 1, 1,
            ↪  -1),
            requires_grad=temp_grad,
        )
        self.phase_gate_proj = nn.Linear((num_heads * head_dim), num_heads, bias=True)

    def forward(
        self,
        q: torch.Tensor,
        k: torch.Tensor,
        inputs: torch.Tensor | None = None,
        output_final_state: bool = False,
        cache: None = None,
        cu_seqlens: None = None,
    ) -> tuple[torch.Tensor, torch.Tensor, torch.Tensor | None]:
        if self.is_softmax:
            q_norm = l2_norm(q)

        phi = rearrange(
            self.phi_proj(
                rearrange(q_norm if self.is_softmax else q, "b t h d -> (b t) h d")
            ),
            "(b t) h d -> b (h d) t",
            b=q.shape[0],
        )
        phi, conv_cache = self.phi_conv1d(
            rearrange(phi, "b d t -> b t d"),
            cache=cache, output_final_state=output_final_state, cu_seqlens=cu_seqlens,
        )
        phi = rearrange(phi,"b t (h d) -> b t h d",h=self.num_heads)
        phase_gate = self.phase_gate_proj(l2_norm(inputs)).sigmoid()
        phi = phi * phase_gate.unsqueeze(-1)
        phi_tilde = torch.cumsum(phi, dim=1)
        qk_phi_tilde = torch.cat([phi_tilde, phi_tilde], dim=2)
        qk_r2 = torch.cat([q, k], dim=2).unflatten(dim=-1, sizes=(-1, 2)).float()
        rotated_qk = torch.stack(
            [
                qk_r2[..., 0] * torch.cos(self.temperature * qk_phi_tilde)
                - qk_r2[..., 1] * torch.sin(self.temperature * qk_phi_tilde),
                qk_r2[..., 1] * torch.cos(self.temperature * qk_phi_tilde)
                + qk_r2[..., 0] * torch.sin(self.temperature * qk_phi_tilde),
            ],
            -1,
        ).flatten(3)
        return torch.split(rotated_qk.type_as(q), q.shape[2], dim=2), conv_cache
```

Figure 9: Selective RoPE in PyTorch.