Mesa-Extrapolation: A Weave Position Encoding Method for Enhanced Extrapolation in LLMs

Xin Ma[1], Yang Liu[2,3]*, Jingjing Liu[2], Xiaoxu Ma[1]
[1]Digital Research Institute, Enn Group, Beijing, China
[2]Institute for AI Industry Research, Tsinghua University, Beijing, China
[3]Shanghai Artificial Intelligence Laboratory, China

NEURAL INFORMATION
PROCESSING SYSTEMS

Introduction

The **extrapolation** problem: as input lengths extend beyond their maximum training limits, Large Language Models (LLMs)’s inference abilities degrade sharply.

Current extrapolation solutions fall into two main categories:

【1】Extended Fine-Tuning: Position Interpolation (PI), Yarn, Unlimiformer, Focused Transformer (FOT), etc.,.

【2】Training-Free Plug-Ins: Streaming-LLM, LM-Infinite, ReRoPE, NTK-based methods, etc.,.

Existing theoretical works on extrapolation in large models: entropy increase, attention logits explode, etc.,.

Theotical Anlysis

Motivation

In a multi-layer neural network, each layer's outputs, a.k.a hidden state values o, become the inputs for the subsequent layer. To maintain stable network behavior, these values must remain within a reasonable range.

We define this observable boundary as the threshold H.

Validating Extrapolation Using Observed Thresholds

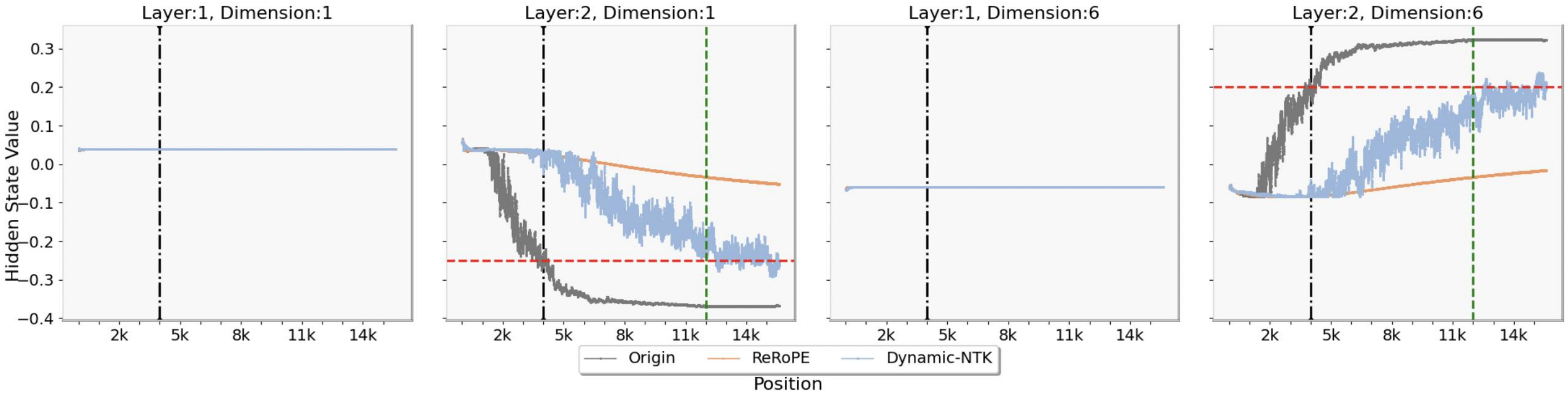
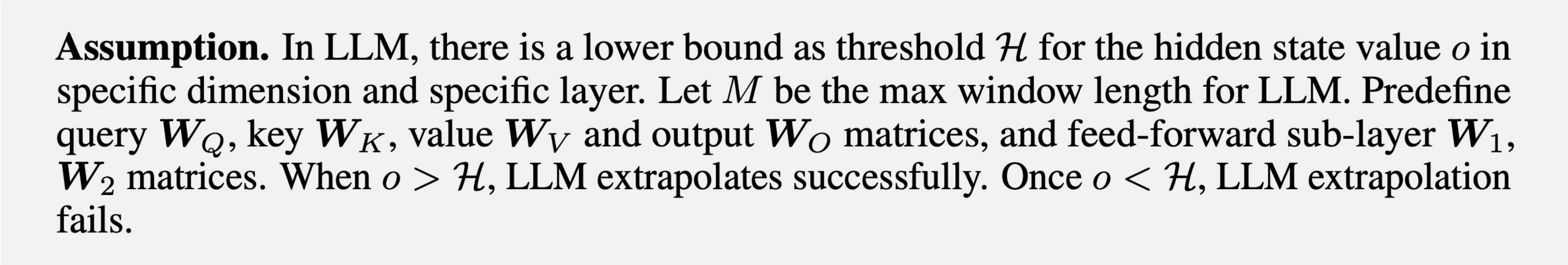
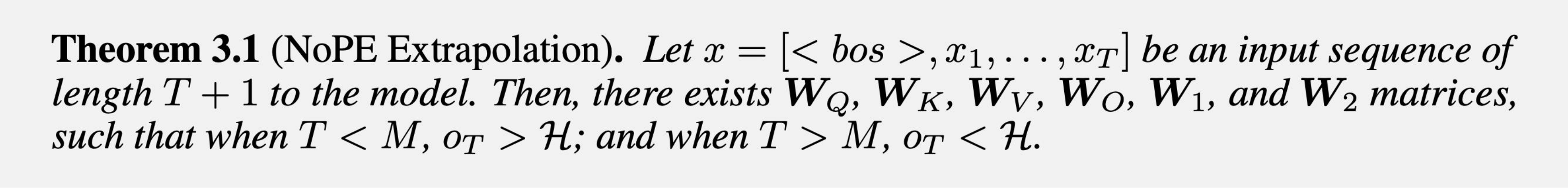


Figure 1: Thresholds for hidden states observed at specific dimensions on LLaMA2-7B-Chat, allowing for extrapolative judgments based on these thresholds. The vertical black dashed line indicate the position of maximum training length of the model. In this case, it is 4k for LLaMA2-7B-Chat model. The hidden state value at this position is designated as the observed threshold and marked with a horizontal red dashed line. When the hidden state value exceeds the red dashed line as the position changes, it signifies that the hidden state value has surpassed the threshold, suggesting a failure in extrapolation after that position.

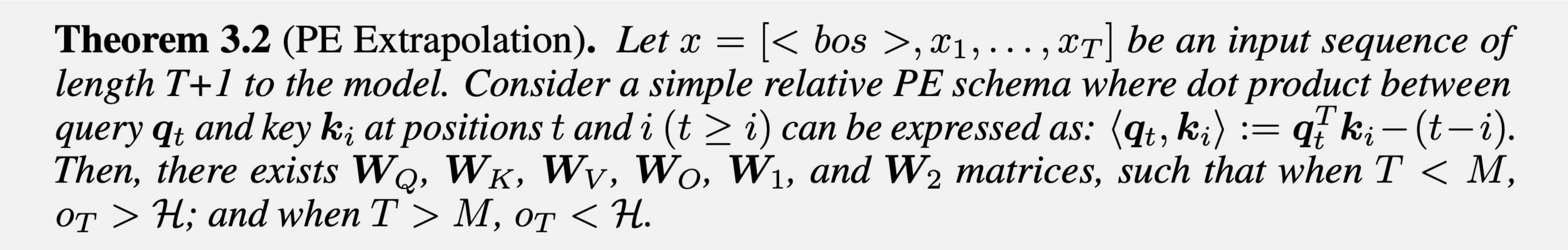
Theotical Anlysis



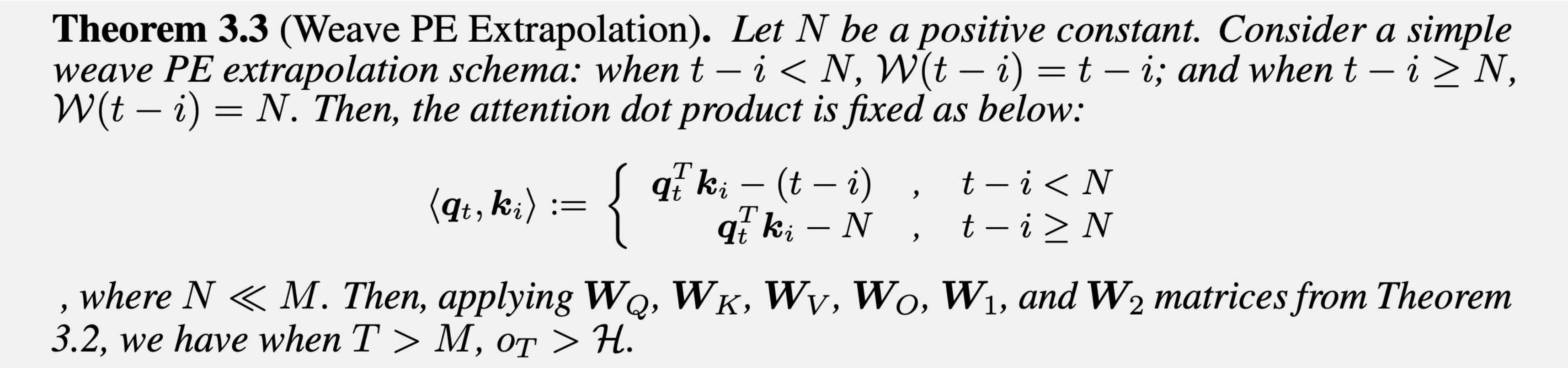
Assumption. In LLM, there is a lower bound as threshold $\mathcal{H}$ for the hidden state value o in specific dimension and specific layer. Let M be the max window length for LLM. Predefine query $\boldsymbol{W}_Q$, key $\boldsymbol{W}_K$, value $\boldsymbol{W}_V$ and output $\boldsymbol{W}_O$ matrices, and feed-forward sub-layer $\boldsymbol{W}_1$, $\boldsymbol{W}_2$ matrices. When $o > \mathcal{H}$, LLM extrapolates successfully. Once $o < \mathcal{H}$, LLM extrapolation fails.



Theorem 3.1 (NoPE Extrapolation). *Let $x = [< bos >, x_1, \ldots, x_T]$ be an input sequence of length $T+1$ to the model. Then, there exists $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_V$, $\boldsymbol{W}_O$, $\boldsymbol{W}_1$, and $\boldsymbol{W}_2$ matrices, such that when $T < M$, $o_T > \mathcal{H}$; and when $T > M$, $o_T < \mathcal{H}$.*



Theorem 3.2 (PE Extrapolation). *Let $x = [< bos >, x_1, \ldots, x_T]$ be an input sequence of length T+1 to the model. Consider a simple relative PE schema where dot product between query $\boldsymbol{q}_t$ and key $\boldsymbol{k}_i$ at positions t and i $(t \geq i)$ can be expressed as: $\langle \boldsymbol{q}_t, \boldsymbol{k}_i \rangle := \boldsymbol{q}_t^T \boldsymbol{k}_i - (t-i)$. Then, there exists $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_V$, $\boldsymbol{W}_O$, $\boldsymbol{W}_1$, and $\boldsymbol{W}_2$ matrices, such that when $T < M$, $o_T > \mathcal{H}$; and when $T > M$, $o_T < \mathcal{H}$.*

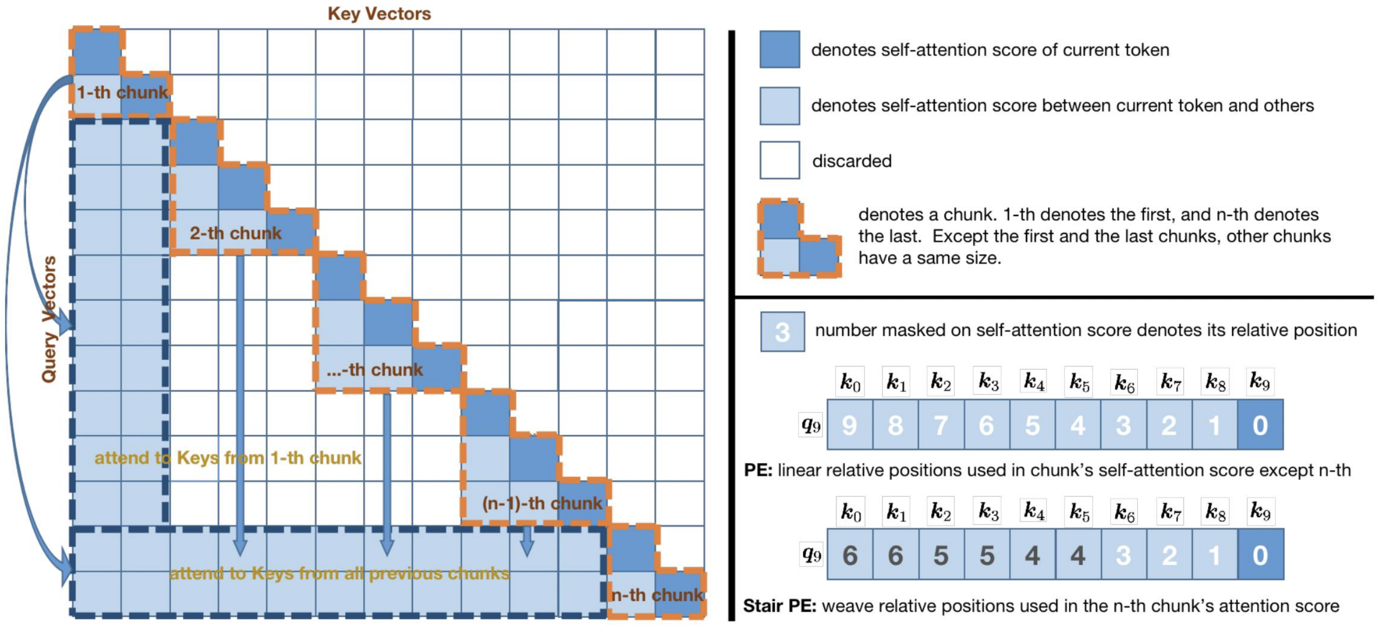


Theorem 3.3 (Weave PE Extrapolation). *Let N be a positive constant. Consider a simple weave PE extrapolation schema: when $t - i < N$, $\mathcal{W}(t-i) = t - i$; and when $t - i \geq N$, $\mathcal{W}(t-i) = N$. Then, the attention dot product is fixed as below:*

$$\langle \boldsymbol{q}_t, \boldsymbol{k}_i \rangle := \begin{cases} \boldsymbol{q}_t^T \boldsymbol{k}_i - (t-i) & , \quad t-i < N \\ \boldsymbol{q}_t^T \boldsymbol{k}_i - N & , \quad t-i \geq N \end{cases}$$

, where $N \ll M$. Then, applying $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, $\boldsymbol{W}_V$, $\boldsymbol{W}_O$, $\boldsymbol{W}_1$, and $\boldsymbol{W}_2$ matrices from Theorem 3.2, we have when $T > M$, $o_T > \mathcal{H}$.

Our Method


Key Vectors
Query Vectors
1-th chunk
2-th chunk
...-th chunk
(n-1)-th chunk
n-th chunk
attend to Keys from 1-th chunk
attend to Keys from all previous chunks
3
k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7 k_8 k_9
q_9 9 8 7 6 5 4 3 2 1 0
k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7 k_8 k_9
q_9 6 6 5 5 4 4 3 2 1 0

denotes self-attention score of current token

denotes self-attention score between current token and others

discarded

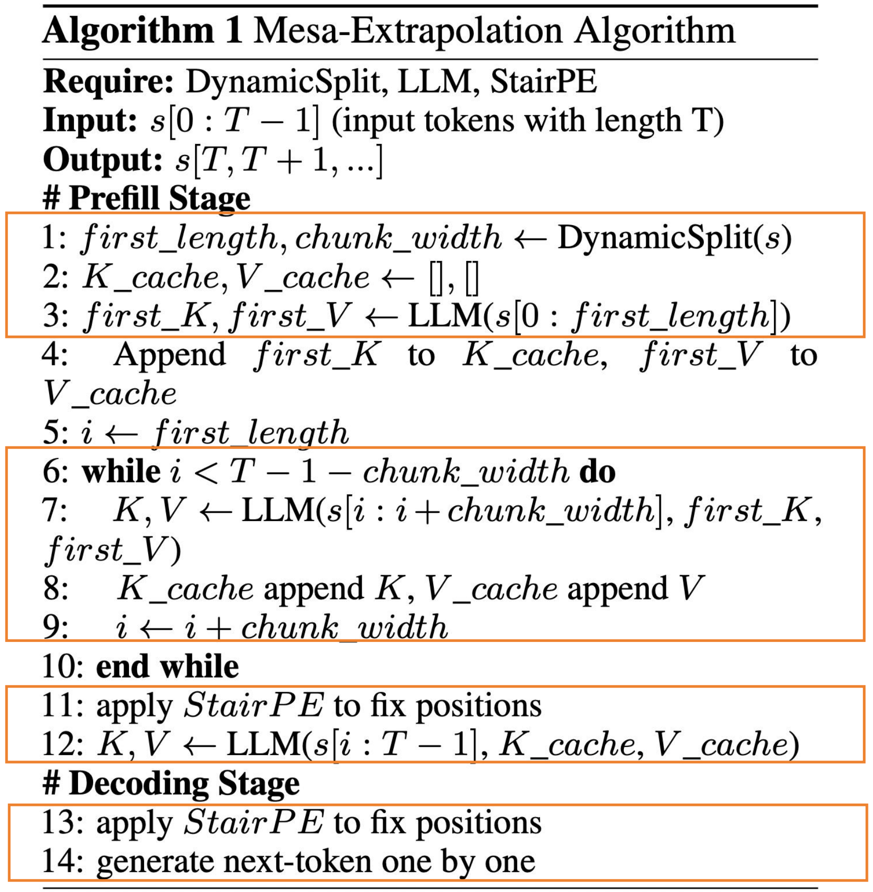
denotes a chunk. 1-th denotes the first, and n-th denotes the last. Except the first and the last chunks, other chunks have a same size.

number masked on self-attention score denotes its relative position

PE: linear relative positions used in chunk's self-attention score except n-th

Stair PE: weave relative positions used in the n-th chunk's attention score

Fig.1 Chunk-based triangular attention matrix, PE and Stair PE. The left figure shows the Chunk-based triangular attention matrix (before SoftMax operation) of Mesa-Extrapolation when an exemplar sequence of length 13 is fed into a LLM. The right figure shows an example of PE and Stair PE. The Stair PE is used to weave the relative position in Mesa-Extrapolation.



Algorithm 1 Mesa-Extrapolation Algorithm

```
Require: DynamicSplit, LLM, StairPE
Input: $s[0 : T-1]$ (input tokens with length T)
Output: $s[T, T+1, ...]$
# Prefill Stage
$first\_length, chunk\_width \leftarrow$ DynamicSplit($s$)
$K\_cache, V\_cache \leftarrow [], []$
$first\_K, first\_V \leftarrow$ LLM($s[0 : first\_length]$)
Append $first\_K$ to $K\_cache$, $first\_V$ to $V\_cache$
$i \leftarrow first\_length$
while $i < T - 1 - chunk\_width$ do
    $K, V \leftarrow$ LLM($s[i : i + chunk\_width]$, $first\_K$, $first\_V$)
    $K\_cache$ append $K$, $V\_cache$ append $V$
    $i \leftarrow i + chunk\_width$
end while
apply $StairPE$ to fix positions
$K, V \leftarrow$ LLM($s[i : T-1]$, $K\_cache$, $V\_cache$)
# Decoding Stage
apply $StairPE$ to fix positions
generate next-token one by one
```

Experiments

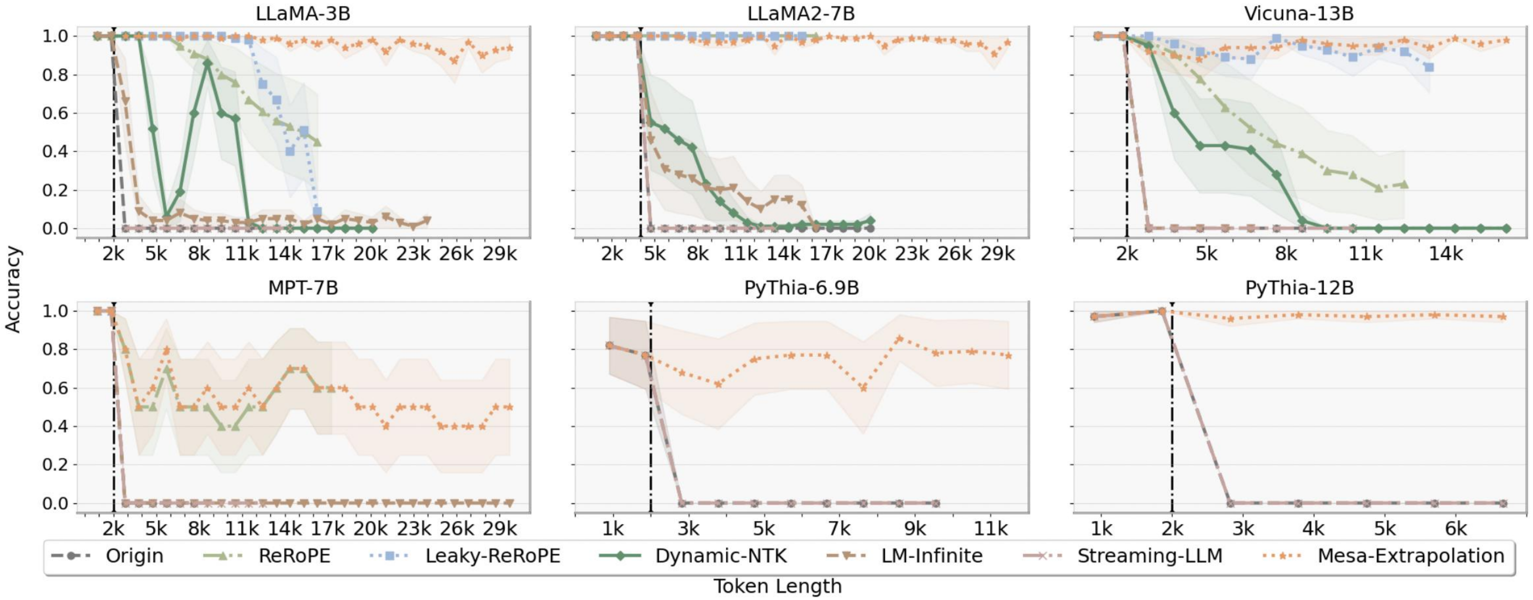


Figure 3: Passkey Retrieval Accuracy for different methods on various LLMs. X-axis represents the input token length, and Y-axis represents the accuracy of password found by LLMs. Different color regions denote the variance value, averaged on 100 samples for each input token length. The black dashed line represent the max training length for LLMs. Some observations: Weave PE-based methods, including ReRoPE, Leaky-ReRoPE, and Mesa-Extrapolation, consistently demonstrate stable extrapolation capabilities even when the input length surpasses the maximum training length. We claim that "early stopping" phenomenon in certain methods is attributed to GPU memory exhaustion under our existing hardware resources.

Experiments

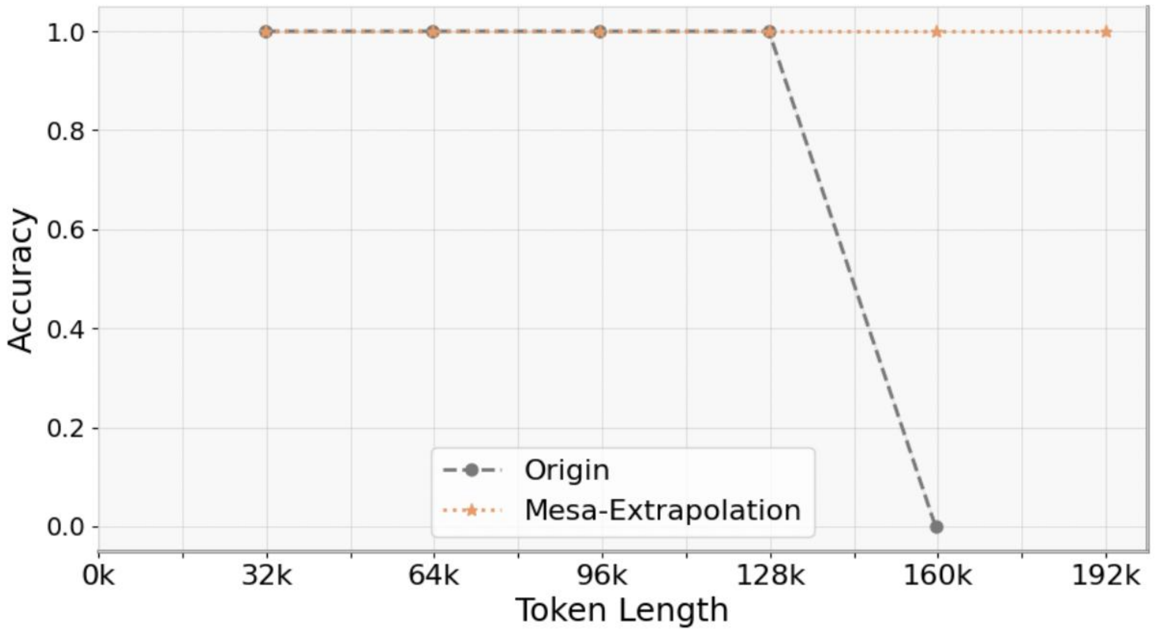
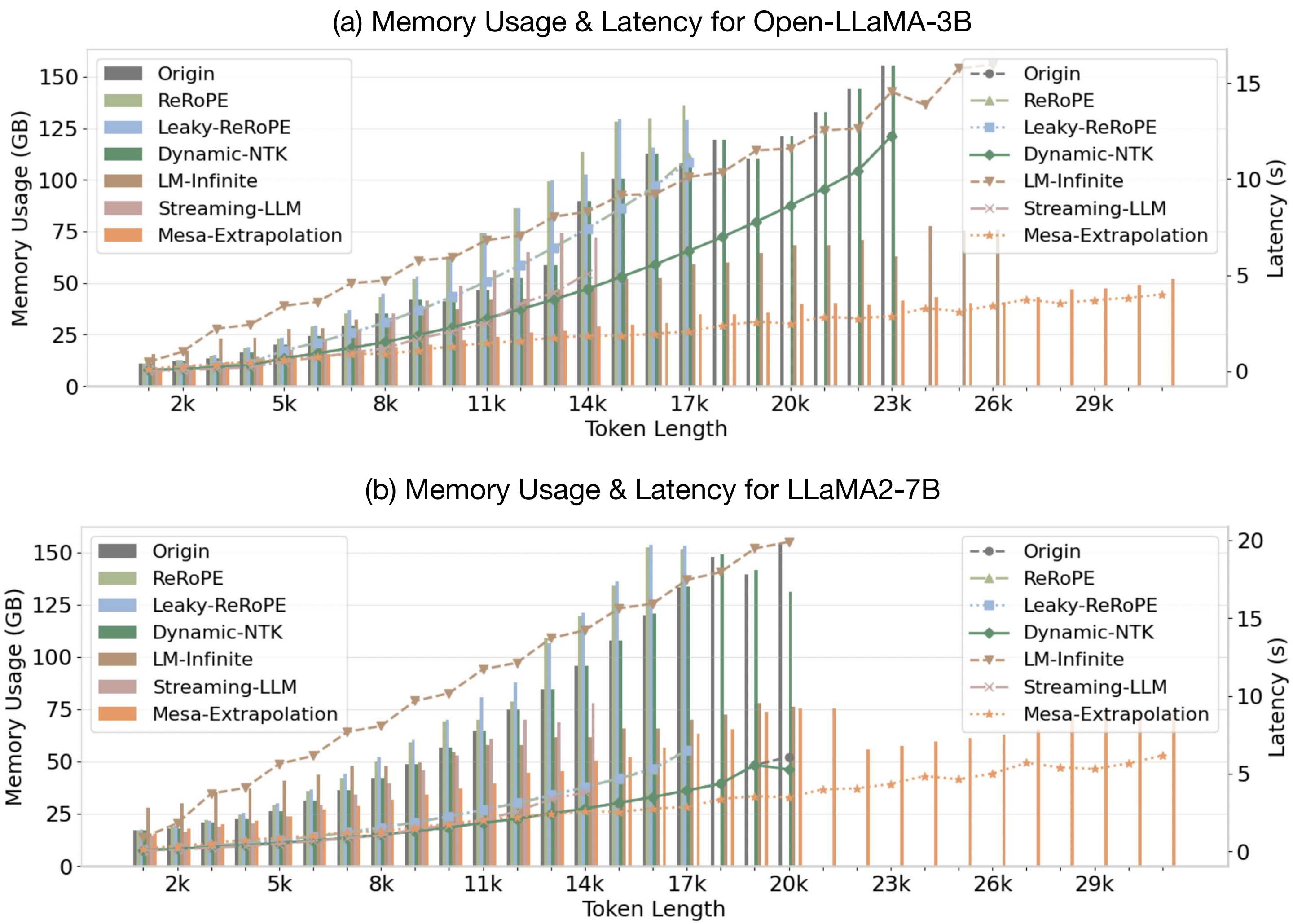


Figure 10: NIAH Task on Phi-3-mini-128k-instruct model using Origin and Mesa-Extrapolation.

Experiments


(a) Memory Usage & Latency for Open-LLaMA-3B

(b) Memory Usage & Latency for LLaMA2-7B

Figure 5: Memory Usage and Decoding Speed Comparison for LLaMA Models: Memory Usage & Latency for LLaMA2-7B

Summary

Key contributions include:

- Theoretical insights on NoPE and PE limitations, and weave PE extrapolation.
- Introducing Mesa-Extrapolation, a practical solution using weave PE and triangular attention
- Empirical validation showing competitive performance, low memory usage, and fast inference

Thanks