Meta LM-Infinite

Lambda-Masking & De-Cluttered Memory

The Paper

- Preprint, still under review
- **Proposes two primary innovations**
  - A **Λ-shaped attention mask** to constrain the number of tokens attended to. This addresses the **issue of high entropy/diluted attention** when attending to too many tokens.
  - **Bounding the relative distances** during attention to a fixed value. This **prevents exploding attention logits** due to unseen long distances.
- **Problems solved**
  - Exploding logits on sequences longer than training data
  - **“Plug-and-play”** solution on multiple Open Source LLMs
- **Results include**
  - Better performance such as passkey retrieval on long sequences
  - Works on sequences far longer than training data

Preprint. Under review.

LM-INFINITE: SIMPLE ON-THE-FLY LENGTH GENERALIZATION FOR LARGE LANGUAGE MODELS

Chi Han[1] **, Qifan Wang**[2]**, Wenhan Xiong**[3]**, Yu Chen**[2]**, Heng Ji**[1]**, Sinong Wang**[3]

[1] University of Illinois Urbana-Champaign, [2] Meta, [3] GenAI Meta
[1]{chihan3,hengji}@illinois.edu,
[23]{wqfcr,xwhan,hugochen,sinongwang}@meta.com

ABSTRACT

In recent years, there have been remarkable advancements in the performance of Transformer-based Large Language Models (LLMs) across various domains. As these LLMs are deployed for increasingly complex tasks, they often face the needs to conduct longer reasoning processes or understanding larger contexts. In these situations, the *length generalization failure* of LLMs on long sequences become more prominent. Most pre-training schemes truncate training sequences to a fixed length (such as 2048 for LLaMa). LLMs often struggle to generate fluent texts, let alone carry out downstream tasks, after longer contexts, even with relative positional encoding which is designed to cope with this problem. Common solutions such as finetuning on longer corpora often involves daunting hardware and time costs and requires careful training process design. To more efficiently leverage the generation capacity of existing LLMs, we theoretically and empirically investigate the main out-of-distribution (OOD) factors contributing to this problem. Inspired by this diagnosis, we propose a simple yet effective solution for on-the-fly length generalization, LM-Infinite, which involves only a Λ-shaped attention mask and a distance limit while requiring no parameter updates or learning. We find it applicable to a variety of LLMs using relative-position encoding methods. LM-Infinite is computational efficient with $O(n)$ time and space, and demonstrates consistent fluency and generation quality to as long as 32k tokens on ArXiv and OpenWebText2 datasets, with 2.72x decoding speedup. On downstream task such as passkey retrieval, it continues to work on inputs much longer than training lengths where vanilla models fail immediately.

1 INTRODUCTION

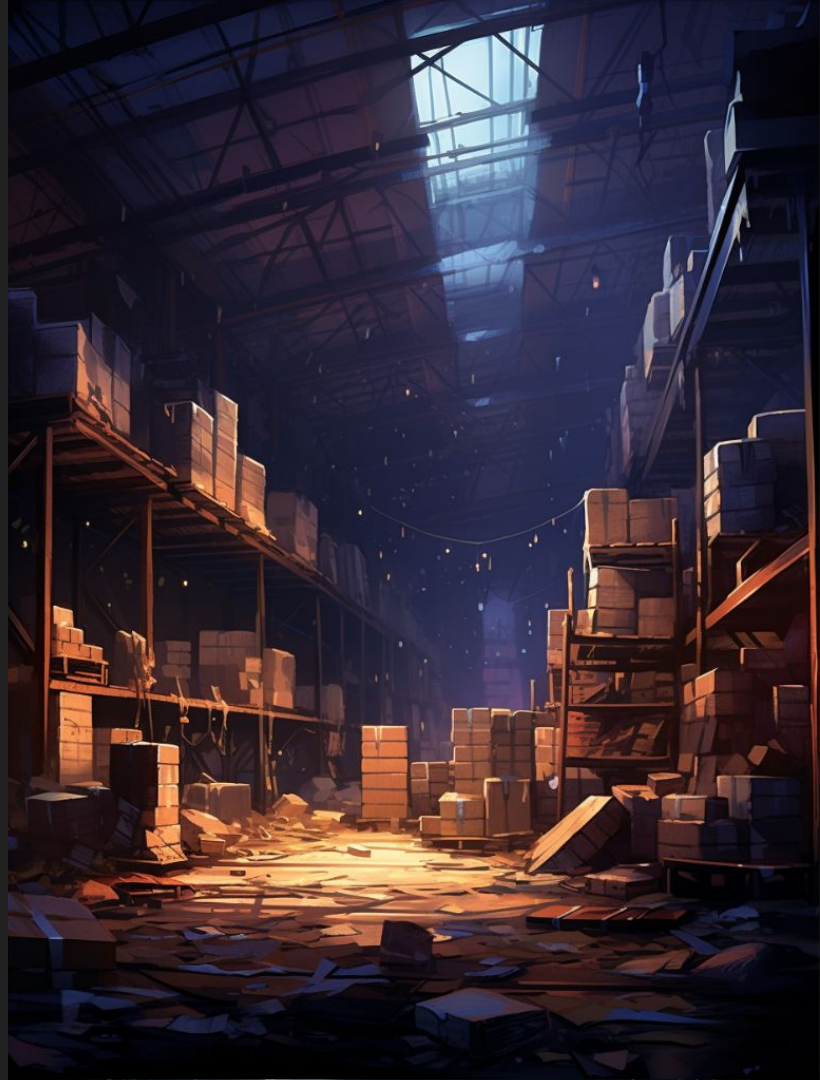
The evolution of Natural Language Generation (NLG) in recent years has been significantly driven by the progress of Large Language Models (LLMs) (Wei et al., 2022a; Kojima et al.; Wei et al., 2022b; Brown et al., 2020; Li et al., 2023b). LLMs have been successfully applied to a wide variety of tasks, demonstrating an impressive ability to understand and generate natural language across different contexts.

However, as LLMs are deployed for more complex tasks such as Document Understanding, Information Extraction and Cross-document Question Answering, they often face the challenge of conducting longer reasoning processes or handling larger volumes of information. This is often reflected in long text sequences, exceeding the typical length in pre-training. However, despite extensive explorations in smaller-scale models (Press et al., 2021; Sun et al., 2022; Chi et al., 2023), current SoTA LLMs still struggle to directly generalize to sequences of unseen lengths. When forced to generate after too long contexts, they either compromise the generation fluency. This challenge is known as *length generalization failures* on LLMs. In most pre-training schemes, to control the exploding time and economic costs with long text lengths, practitioners have to bound training sequences, such as 2048 tokens for LLaMA for 4096 for LLaMA-2. When there is a gap between training and inference lengths, LLMs fail to recognize the input and start to generate gibberish,

*Work performed as an intern in Meta GenAI.

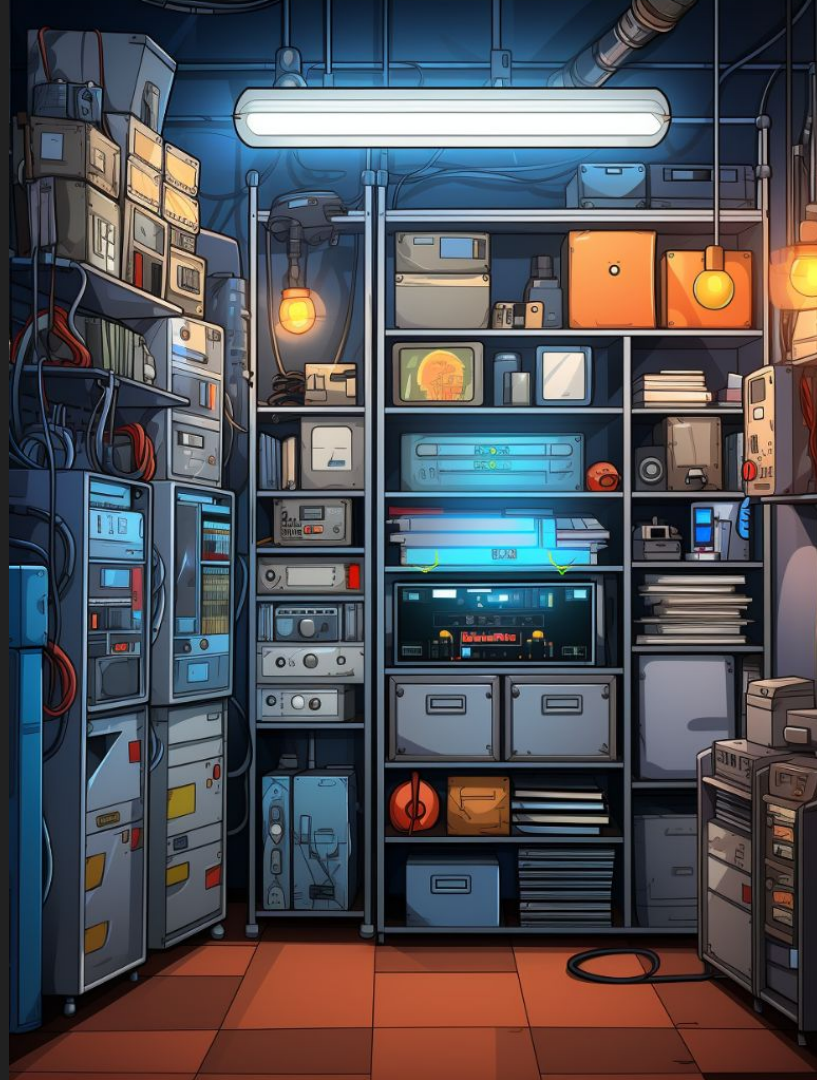
Cluttered Warehouse

- Large context windows are like **enormous warehouses**
- Lots and lots of **tokens to keep track of**
- **Mathematical complexity** of this volume becomes untenable
- **Similar to human working memory**
  - **Cluttered and overloaded working memory impairs reasoning**
  - Constrained, cleaner working memory yields better performance
  - Basically forces LLM to do garbage collection



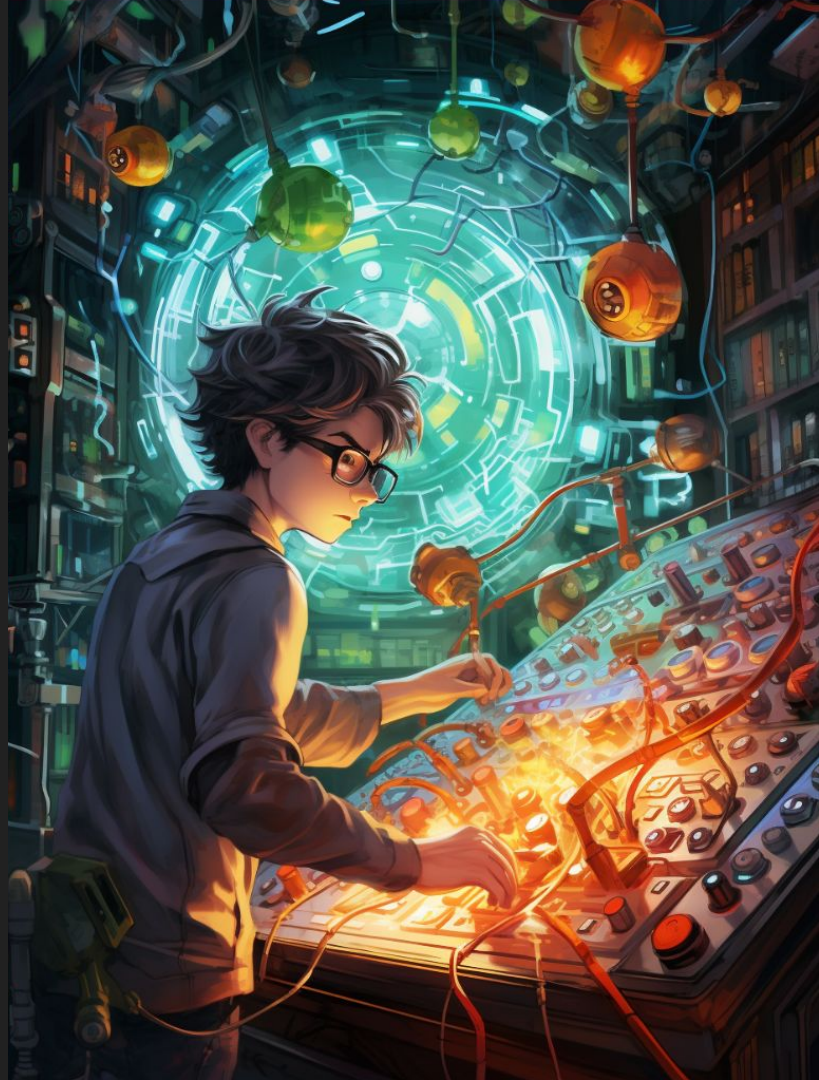
Tidy Storage Closet

- Less to keep track of
- Mathematically simpler
- **Which one would you rather be responsible for?**
- **Prevents dilution and degeneration**
- Pretty much impossible to get lost in this cleaner space



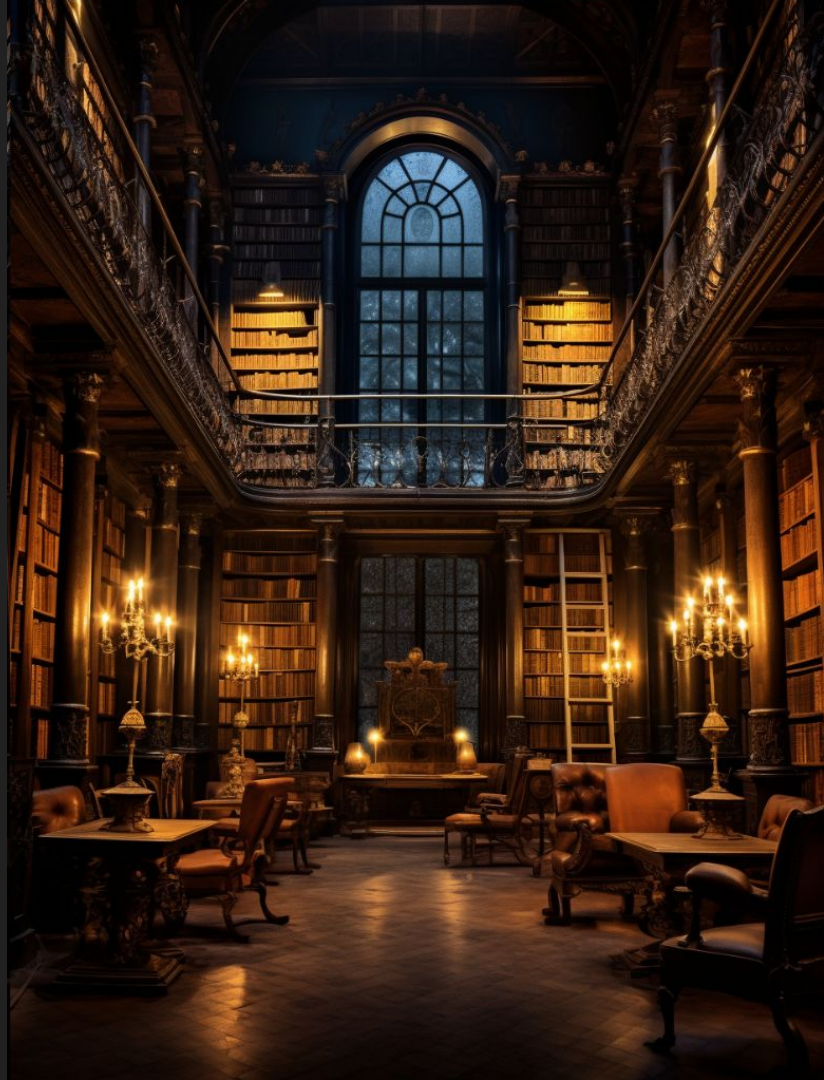
Easy Tweaks

- **Tested on several open source LLMs**
  - LLaMA, GPT-J, and MPT-7B
- **Utilizes existing mechanisms**
  - Just changes the algorithm a little bit
- **Analogous to garbage collection**
  - Frees up memory (attention) by masking unused tokens
  - Forces prioritization, constranes size of memory
  - Not a perfect analogy, but sensible enough
  - **The Λ-mask privileges retaining the newer local context while masking away older tokens**
  - **Garbage collectors prioritize collecting older unreferenced data first**
  - **Both aim to create free space and reduce clutter in memory without losing critical information**



So what?

- **Might ultimately change the way we interact with LLMs**
  - Better attention mechanisms
  - Bigger windows
  - Combine with One Billion Tokens
  - **Might invalidate vector search**
- **Examples**
  - **Writing Fiction:** Keep track of millions of words across many books
  - **Scientific Research:** Survey and correlate thousands of papers at once
  - **Enterprise Business:** Search thousands of company documents simultaneously
- **TLDR**
  - Efficient working memory akin to garbage collection
  - Better performance, especially on long contexts



Thank you