Thus Spake Long-Context Large Language Model

Xiaoran Liu[1,2]*, Ruixiao Li[2]*, Mianqiu Huang[2]*, Zhigeng Liu[2]*, Yuerong Song[2]*,
Qipeng Guo[1,4], Siyang He[2], Qiqi Wang[2], Linlin Li[3], Qun Liu[3],
Yaqian Zhou[2], Xuanjing Huang[2], Xipeng Qiu[1,2,4]†
[1]Shanghai AI Lab, [2]School of Computer Science Fudan University,
[3]Huawei Noah's Ark Lab, [4]Shanghai Innovation Institute
xrliu24@m.fudan.edu.cn, guoqipeng@pjlab.org.cn, xpqiu@fudan.edu.cn

arXiv:submit/6146238 [cs.CL] 22 Jan 2025

Abstract

Long context is an important topic in Natural Language Processing (NLP), running through the development of NLP architectures, and offers immense opportunities for Large Language Models (LLMs) giving LLMs the lifelong learning potential akin to humans. Unfortunately, the pursuit of a long context is accompanied by numerous obstacles. Nevertheless, long context remains a core competitive advantage for LLMs. In the past two years, the context length of LLMs has achieved a breakthrough extension to millions of tokens. Moreover, the research on long-context LLMs has expanded from length extrapolation to a comprehensive focus on architecture, infrastructure, training, and evaluation technologies.

Inspired by the symphonic poem, *Thus Spake Zarathustra*, we draw an analogy between the journey of extending the context of LLM and the attempts of humans to transcend its mortality. In this survey, We will illustrate how LLM struggles between the tremendous need for a longer context and its equal need to accept the fact that it is ultimately finite. To achieve this, we give a global picture of the lifecycle of long-context LLMs from four perspectives: architecture, infrastructure, training, and evaluation, showcasing the full spectrum of long-context technologies. At the end of this survey, we will present 10 unanswered questions currently faced by long-context LLMs. We hope this survey can serve as a systematic introduction to the research on long-context LLMs.

Acknowledgement

This work is supported by the project cooperated with Huawei Noah's Ark Lab, *Research on New and Efficient Architectures for Large-scale Language Models*, under the direction of Prof Qiu and Prof Guo in OpenMOSS FNLP as well as InternLM Team.

If possible, I would like to dedicate this work to Prof Guo, who has worked hard over the past six months, to the memorable year with Boss Hang and Master Zhang in AI Lab, to the unforgettable moments in XinKingBo, and to the invaluable recognition from Prof Qiu.

Due to the authors' limited knowledge, this survey may contain omissions. We welcome constructive comments from readers. We will carefully consider these suggestions and release a revised version in two to three months.

* Equal contribution.
† Corresponding Author.

1 Introduction

Research on long-context capability has been an important topic in Natural Language Processing (NLP), reflected in the evolutionary trajectory of mainstream architectures. This evolution shows a consistent progression toward increasing context length, from the Bag-of-Word models (Har) with no concept of context to CNNs (LeCun et al., 1995) with local receptive fields, then to LSTMs (Schmidhuber et al., 1997) characterized with an explicit long short-term memory, and currently to Transformer featured with modeling long-range dependencies (Vaswani et al., 2017), as well as the recent discussions on the SSM-Mamba series (Gu et al., 2020; Gu & Dao, 2023; Dao & Gu, 2024) that challenges the dominance of Transformers from the perspective of history storage. Researchers hope models, especially the Large Language Model (LLM) (OpenAI, 2023a; Sun et al., 2024c; Reid et al., 2024; Meta, 2024a), can possess life-long context, rather than being limited by a fixed window size.

As shown in Figure 1, in a 1k context, LLM may only understand a short fairy tale. In a 4k context, the reading comprehension may be limited to an arXiv paper (Shaham et al., 2022). In a 32k to 128k context, LLM may process a hundreds-page detective novel in its entirety and successfully infer the identity of the murderer (Xu et al., 2024f; Wang et al., 2024b). When the context length extends to 512k, even a novel as lengthy as Ulysses or a novel series could be input and understood as a whole (Jacobs et al., 2023). When the context length reaches 2M, the model may learn new knowledge through many-shot long In-Context Learning (ICL) (Agarwal et al., 2024) or acquire a new language via vocabulary and grammar books (Reid et al., 2024). If the context length becomes infinite, LLM may possess life-long learning capabilities, which can fundamentally change the existing paradigms of training (Sun et al., 2024f; Lin et al., 2024a).

Unfortunately, as context length increases, researchers also face various obstacles. From an architectural perspective, the context length of mainstream Transformer architectures is limited not only by the pre-training window size (Press et al., 2022; Chen et al., 2023b) but also by the memory and computational overhead of the Key-Value (KV) cache (Kwon et al., 2023; Xiao et al., 2024c). From an infrastructural perspective, longer contexts result in greater memory pressure and lower throughput (Chen et al., 2024e; Kwon et al., 2023). From a training perspective, long-context datasets face challenges in both quantity and quality (Lv et al., 2024a; Gao et al., 2024d). From an evaluation perspective, increasing context length reveals more potential problems in LLMs (Agarwal et al., 2024; Hsieh et al., 2024a), leading to higher requirements for LLM performance (Xu et al., 2024f; Zhang et al., 2023d).

However, since the emergence of LLM, long-context capabilities remain one of the most rapidly developing areas and constitute a core competition point (Anthropic, 2024a; Cai et al., 2024c; Meta, 2024a), as shown in Figure 2. From April 2023 to February 2024, the context

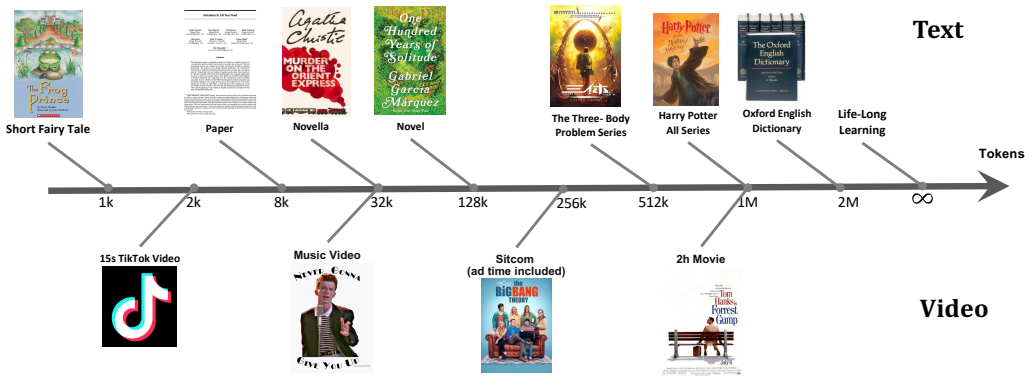

Figure 1: Context window comparison across text and video: The timeline illustrates the token counts. Text uses the tiktoken (OpenAI, 2023b) tokenizer, while video uses the Qwen2-VL (Wang et al., 2024j) tokenizer.

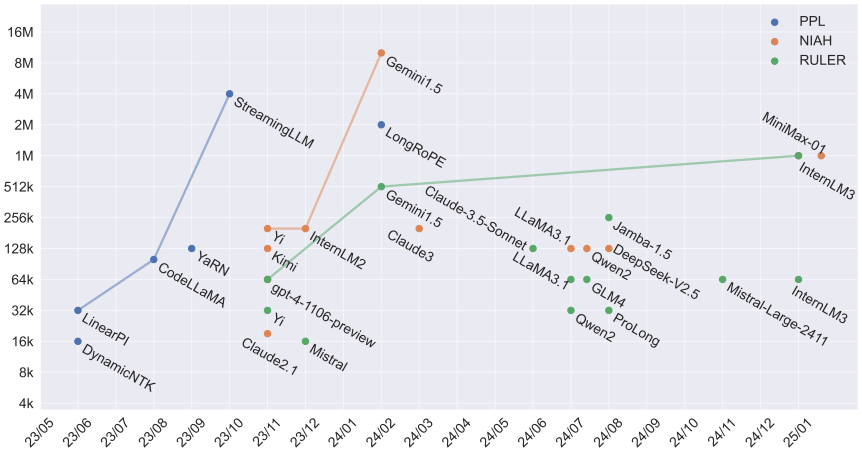

Figure 2: Long-context performance of various LLMs across multiple benchmarks, perplexity (PPL) (Press et al., 2022), NIAH (Kamradt, 2023), and RULER (Hsieh et al., 2024a). The horizontal axis represents the release time, while the vertical axis indicates the effective context length achieved by the LLMs on the corresponding task. The line associated with each task represents the state-of-the-art performance at a given point in time.

length of open-source LLMs has grown from an initial 2k (Touvron et al., 2023a) to 2M (Ding et al., 2024). In this process, some surveys concentrate on particular aspects (Huang et al., 2023; Zhao et al., 2023b; Pawar et al., 2024), particularly developments in architectural design, while other technique reports focus on summarizing the life-cycle of a specific long-context LLM (ChatGLM, 2024; Gao et al., 2024d), from data construction to context extension and to performance evaluation. Currently, there is a lack of a comprehensive survey that presents the full life-cycle of long-context LLMs from architecture, infrastructure, training, and evaluation, showing the global picture of long-context technology.

Inspired by *Thus Spake Zarathustra*, the symphonic poem of the German composer Richard Strauss, we draw an analogy between the attempts of LLMs to extend their context lengths and the attempts of humans to transcend their mortality. On the journey of extending the context length of LLMs, researchers continuously challenge the boundaries of context through optimizations in architecture, infrastructure, and training, much like *the struggle between man's tremendous need for immortality and his equal need to accept the fact that he is mortal*[1]. As shown in Figure 3, this survey comprehensively introduces the life-cycle of long-context LLMs from four perspectives: **architecture**, **infrastructure**, **training**, and **evaluation**.

- Sections 2 to 5 focus on the architectural aspect, discussing the enhancement of Transformer in length extrapolation, KV cache optimization as well as memory management, and the innovation to defeat Transformer by long-context researchers.
- Sections 6 and 7 address infrastructure considerations, detailing optimizations for long context in the training and inference phases of Transformer-based LLMs.
- Sections 8 to 10 introduce the training methods in three corresponding training stages for long-context LLMs, pre-training, post-training, and multi-modal training, particularly for long-context Multi-modal LLM (MLLM).
- In Section 11, we will discuss the long-context evaluation. In Section 12, we will outline 10 unanswered questions that long-context LLMs still face as a conclusion.

[1]From *Thus Spake Richard Strauss* by Leonard Bernstein in Young People's Concert. https://leonardbernstein.com/lectures/television-scripts/young-peoples-concerts/thus-spake-richard-strauss

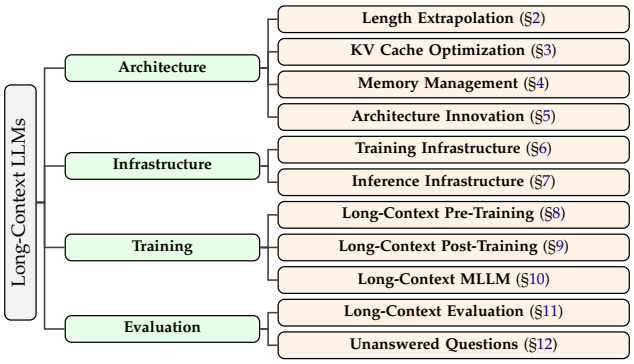

Figure 3: An overview of *Thus Spake Long-Context Large Language Model*.

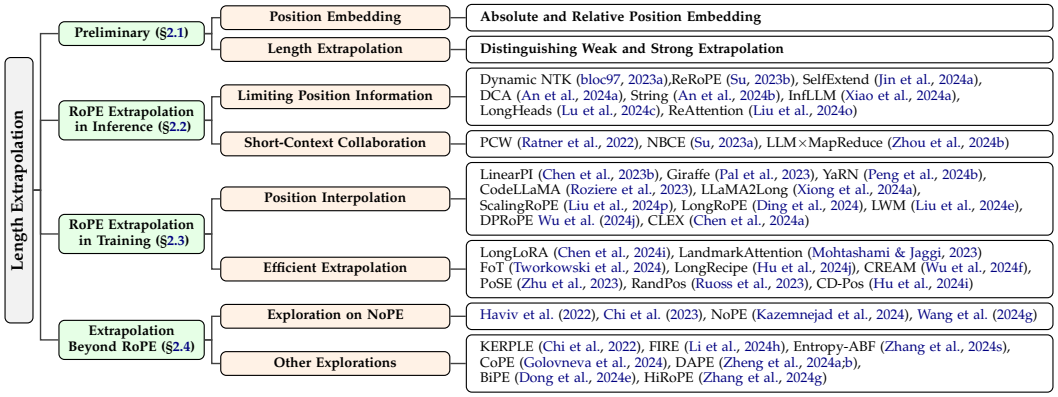

Figure 4: An overview of length extrapolation of long-context LLMs.

2 Length Extrapolation

In this section, we start the journey of extending the context length of LLMs with length extrapolation, the foundation of long-context LLMs, as shown in Figure 4.

- In §2.1, we start with some preliminary knowledge, including ***position embedding*** and ***the definition of length extrapolation***. Then we focus on the length extrapolation based on the widely-used RoPE (Su et al., 2024).
- In the inference stage, as discussed in §2.2, the extrapolation is based on ***limiting position information*** including NTK (bloc97, 2023a), ReRoPE (Su, 2023b) and DCA (An et al., 2024a) or ***short-context collaboration*** like PCW (Ratner et al., 2022).
- In the training stage, as discussed in §2.3, apart from the classical ***position interpolation*** methods such as LinearPI (Chen et al., 2023b) and YaRN (Peng et al., 2024b), we highlight the discussion of ***extrapolation mechanism*** (Liu et al., 2024p; Men et al., 2024) and ***efficient extrapolation*** (Chen et al., 2024i; Zhu et al., 2023).
- In §2.4, we will add more discussion beyond RoPE, including ***NoPE*** (Kazemnejad et al., 2024), other position embeddings (Golovneva et al., 2024; Dong et al., 2024e) and ***attention entropy*** (Han et al., 2024; Zhang et al., 2024s)

2.1 Preliminary

2.1.1 Position Embedding

First introduced in Vaswani et al. (2017), position embedding is a key mechanism for encoding positional information in contexts, and remains fundamental to modern LLMs' ability

to process long-context. The evolution of position embedding begins with absolute position embedding (Vaswani et al., 2017), namely embedding based on the token indices, and follows by the emergence of relative position embedding, namely embedding based on the token distance, such as Shaw et al. (2018), T5 (Raffel et al., 2020), TENER (Yan et al., 2019) and XLNET (Dai et al., 2019). However, those embeddings face trade-offs between performance and computational efficiency. Later, RoPE (Su et al., 2024) is proposed, achieving relative position embedding through absolute position embedding, combining the advantages of both approaches and thus becoming a significant academic interest (Chowdhery et al., 2023; Touvron et al., 2023a;b; Sun et al., 2024c; Chen et al., 2023b; bloc97, 2023a).

RoPE (Su et al., 2024) introduces positional information into self-attention computation through rotary transformations. Given a position index t and an embedding vector $\boldsymbol{x} = [x_0, x_1, ...x_{d-1}]^T$, where d is the attention head dimension, RoPE defines a complex function:

$$\boldsymbol{A}_{m,n} = \underbrace{\boldsymbol{x}_m \boldsymbol{W}_Q \boldsymbol{R}^d_{\Theta,m-n} \boldsymbol{W}_K^T \boldsymbol{x}_n^T}_{\text{relative position embedding}} = \underbrace{\boldsymbol{x}_m \boldsymbol{W}_Q \boldsymbol{R}^d_{\Theta,m} \left(\boldsymbol{x}_n \boldsymbol{W}_K \boldsymbol{R}^d_{\Theta,n}\right)^T}_{\text{absolute position embedding}}$$

$$\boldsymbol{R}^d_{\Theta,t} = \begin{bmatrix} \cos t\theta_0 & -\sin t\theta_0 & \cdots & 0 & 0 \\ \sin t\theta_0 & \cos t\theta_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \cos t\theta_{d/2-1} & -\sin t\theta_{d/2-1} \\ 0 & 0 & \cdots & \sin t\theta_{d/2-1} & \cos t\theta_{d/2-1} \end{bmatrix}, \tag{1}$$

where $\theta_j = \beta^{-2j/d}$, with a typical value of rotary base $\beta = 10000$.

RoPE has several significant advantages. First, RoPE has solid mathematical foundations with theoretical guarantees (Su et al., 2024). Second, RoPE maintains low computational complexity and eliminates the necessity for storing any position embedding matrices (Su et al., 2024; Touvron et al., 2023a; Chowdhery et al., 2023). Third, RoPE can seamlessly integrate with many attention variants, demonstrating excellent compatibility (Su et al., 2024). Finally, through the following improvements, RoPE has shown strong length extrapolation capabilities (bloc97, 2023a; Liu et al., 2024p). Given these favorable properties of RoPE, many LLMs adopt RoPE as their position embedding (Dubey et al., 2024; GLM et al., 2024; Wang et al., 2024j; Young et al., 2024; Cai et al., 2024c).

2.1.2 Length Extrapolation

In Transformer-XL (Dai et al., 2019), discover the standard Transformer's limitations in handling sequences longer than its training length. ALiBi (Press et al., 2022) later formalizes this as **length extrapolation** or **length generalization**, a model's capacity to maintain performance when processing longer sequences during inference than during training. Before the widespread adoption of RoPE-based linear interpolation, early approaches to length extrapolation include improvements in position embedding and sliding window mechanism (Press et al., 2022; Sun et al., 2022b; Ratner et al., 2022).

For example, ALiBi (Press et al., 2022; Yang et al., 2023) introduces fixed attention biases that scale linearly with relative positional information, showing promising results in contexts beyond training length. Later, xPos (Sun et al., 2022b; 2023) addresses the length extrapolation problem by incorporating exponential decay in attention computation and proposing BCA, a windowed attention mechanism similar to a sliding window. After LLM emerges, the sliding window mechanism is first been used for the earliest length extrapolation attempts (Bai et al., 2023a; Jiang et al., 2023a). Besides, more sophisticated sliding window variants are proposed. For example, LongNet (Ding et al., 2023) achieves length extrapolation to 1B tokens by dilated sliding window attention. Subsequently, LM-Infinite (Han et al., 2024) and StreamingLLM (Xiao et al., 2024c) introduce Λ-shaped masks and attention sinks respectively. These two methods preserve information from global initial tokens and local window tokens, implementing attention window truncation to reduce computational complexity while maintaining LLM's performance.

Distinguishing Weak and Strong Extrapolation It is essential to distinguish two types of extrapolation capabilities, weak extrapolation and strong extrapolation. ***Weak extrapolation*** refers to maintaining perplexity across varied context lengths, while ***strong extrapolation*** indicates the ability to maintain performance on actual long-context understanding and processing tasks. These capabilities can be delineated by examining which tasks maintain consistent performance across different context lengths. For instance, StreamingLLM (Xiao et al., 2024c) demonstrates effective weak extrapolation in perplexity but does not guarantee equivalent performance in practical long-context tasks, as evaluated by benchmarks including NIAH (Kamradt, 2023) and RULER (Hsieh et al., 2024a). The conflict of perplexity between its failure to reflect practical context length and its wide application in the long-context research will be further analyzed in **Q3** in Section12.

This distinction is crucial, as many length extrapolation works focus only on weak extrapolation (Han et al., 2024; Xiao et al., 2024c; Ding et al., 2023). The following discussion focuses on strong extrapolation. Given LLMs' predominant use of RoPE, we first explore the extrapolation of RoPE-based LLMs. Based on implementation stages, these methods can be categorized into inference-time and training-time extrapolation.

2.2 RoPE Extrapolation in Inference

At inference time, there are two feasible approaches for enabling LLMs to comprehend longer context lengths. The first approach involves constraining position embeddings during the processing of extended contexts, and the second approach implements segmented understanding where the model processes long contexts in chunks and integrates understanding across these segments.

2.2.1 Limiting Position Information

In RoPE (Su et al., 2024), positional information is represented through trigonometric functions of the product of index and rotary angle. To maintain this product within pre-training bounds as indices increase, approaches including limiting index growth or reducing rotary angles are proposed. Fixed or dynamic NTK methods (bloc97, 2023b;a) achieve plug-and-play length extrapolation by adjusting RoPE's rotary base and have been widely adopted, while more extrapolation works in inference focus on index limitation.

ReRoPE (Su, 2023b) and SelfExtend (Jin et al., 2024a) explicitly set relative position upper bounds in RoPE to constrain positional information within pre-training ranges. Similarly, InfLLM (Xiao et al., 2024a) and LongHeads (Lu et al., 2024c) enable training-free processing of ultra-long sequences through block-level context storage, focusing attention on crucial blocks at the beginning, end, and middle of input text. ReAttention (Liu et al., 2024o) implements customized operators for fine-grained KV cache retrieval across the full context, enabling plug-and-play context window expansion by at least 100 times. DCA (An et al., 2024a) innovatively decomposes long sequence attention computation into intra-block, adjacent-block, and non-adjacent block components for more efficient long text processing, while String (An et al., 2024b) further simplifies this design and improves performance.

2.2.2 Short-context Collaboration

Short-context Collaboration refers to a series of extrapolation methods that process long texts by splitting them into shorter segments and synthesizing the results. PCW (Ratner et al., 2022) ensures all processing remains within pre-training length limits by dividing sequences into multiple context segments and one task sequence. NBCE (Su, 2023a) applies Naive Bayes principles to achieve length extrapolation through independent processing of context segments with prompts. XL3M (Wang et al., 2024k) introduces a training-free framework handling long contexts through segmented inference, while LLM×MapReduce (Zhou et al., 2024b) adopts distributed computing concepts, processing text blocks across GPUs with specialized communication structures. Additionally, LongAgent (Zhao et al., 2024a), an extrapolation method in training, also employs a similar approach by introducing multi-agent collaboration, where multiple agents cooperate to process long contexts.

2.3 RoPE Extrapolation in Training

2.3.1 Position Interpolation

Beyond extrapolation methods in inference, researchers propose numerous approaches in training that focus on leveraging short-context positional information for longer contexts through position interpolation (Liu et al., 2024p; Xiong et al., 2024a). These methods similarly address either index adjustment or rotary base scaling.

For index adjustment, LinearPI (Chen et al., 2023b) first introduces linear scaling of position indices through a scaling factor to extend context length. However, it remains limited by training length and neglects feature differences across RoPE's query and key vectors' dimensions. YaRN (Peng et al., 2024b) subsequently implements dynamic scaling in middle dimensions while maintaining no interpolation in low dimensions and full interpolation in high dimensions, achieving 128k length extrapolation with 64k training. YaRN gains wide adoption in subsequent LLMs like LLaMA3.1 (Dubey et al., 2024). Similarly, Giraffe (Pal et al., 2023) achieves extrapolation by preserving high-frequency rotations while suppressing low-frequency ones. Additionally, LongRoPE (Ding et al., 2024) employs progressive search-based non-uniform interpolation to achieve 2M context length with 256k training.

On the other hand, many models adopt enlarged rotary angles combined with longer training lengths (Roziere et al., 2023; Xiong et al., 2024a). This approach is widely adopted in current LLMs (Cai et al., 2024c; Young et al., 2024; ChatGLM, 2024) to achieve long contexts. LWM (Liu et al., 2024e) implements multi-stage scaling, gradually increasing both the rotary angle base and fine-tuning length. However, these works make specific attempts on certain context lengths and rotary bases without thoroughly investigating the extrapolation mechanism of RoPE-based LLMs. Apart from the search for mechanism, DPRoPE (Wu et al., 2024j) explores optimizing RoPE rotary angle distributions to enhance extrapolation capabilities and CLEX (Chen et al., 2024a) introduces neural ordinary differential equations to model continuous scaling of position embedding.

2.3.2 Scaling Laws

As previously discussed, the extrapolation mechanism of RoPE-based LLMs remains a crucial question in length extrapolation research. The keys to this question are the ***periodicity*** and ***monotonicity*** of trigonometric functions (Peng et al., 2024b; Liu et al., 2024p; Men et al., 2024). YaRN (Peng et al., 2024b) first mentions the relationship between the RoPE-based extrapolation and the periodicity. Furthermore, ScalingRoPE (Liu et al., 2024p) identifies a critical dimension d_{extra}, decided by the pre-training context length T_{train} and original rotary base β, that determines the LLM's extrapolation limit, as shown in Equation 2.

$$d_{\text{extra}} = 2\left\lceil \frac{d}{2} \log_{\beta} \frac{T_{\text{train}}}{2\pi} \right\rceil. \tag{2}$$

For dimensions before the critical dimension, their position embedding $\sin(\theta t), \cos(\theta t)$ have already experienced a complete period in pre-training and will not be out-of-distribution (OOD) in extrapolation. However, dimensions beyond that will fail to extrapolate when the product of the rotary angle and position index exceeds the range the LLM pre-trained in. Since rotary angles in RoPE are arranged exponentially (Su et al., 2024), the rotary angle at the critical dimension experiences the least shrinkage in base scaling. Consequently, the position embedding at this dimension will first be OOD, making its period serve as the upper bound for extrapolation, T_{extra}, as shown in Equation 3.

$$T_{\text{extra}} = 2\pi \cdot \beta^{\frac{d_{\text{extra}}}{d}} = 2\pi \cdot \beta^{\left\lceil \frac{d}{2} \log_{10000} \frac{T_{\text{train}}}{2\pi} \right\rceil \cdot \frac{2}{d}}. \tag{3}$$

Liu et al. (2024p) reveals a part of the extrapolation mechanism in RoPE-based LLM, that RoPE's extrapolation is representing position information in a longer context using that previously learned in short-context pre-training. However, forcing LLM to learn more position information in fine-tuning, such as reducing the rotary base, is inappropriate (Men et al., 2024). Men et al. (2024) proves that reducing rotary bases undermines contextual information modeling because it disrupts the original patterns and overlooks the second

feature, monotonicity. The $\cos(\theta t)$ maintains monotonicity locally, reflecting relative distance. A sufficiently smaller base can prevent position embedding from OOD based on periodicity, but this sacrifices monotonicity, limiting LLMs to perceiving local semantics and performing poorly on generation and ICL tasks (Liu et al., 2024p; Men et al., 2024), showing only weak extrapolation. This reveals a contradiction in RoPE, that ***dimensions with monotonicity perceivable of long dependencies are overfitted to pre-training context and cannot extrapolate, while dimensions capable of extrapolation lose monotonicity and cannot perceive long contexts***, which will be further analyzed in **Q2** in Section12.

Although Liu et al. (2024p) makes a mistake on the second part, it still has a guiding significance for length extrapolation (Cai et al., 2024c; Apple, 2024). For instance, by finding the inverse function of Equation 3, we can determine the minimum necessary rotary base for supporting a specific context length T_{extra}. Compared to the linear relationship between rotary base β and T_{extra} in Hugging Face's default dynamic NTK implementation , Equation 4 demonstrates a power law which accounts for the extrapolation limit in the NTK approach.

$$\beta = \left(\frac{T_{\text{extra}}}{2\pi} \right)^{\frac{d}{d_{\text{extra}}}}. \tag{4}$$

2.3.3 Efficient Extrapolation

Length extrapolation methods in training also consider achieving extrapolation effects with fewer computational resources, known as efficient extrapolation (Chen et al., 2023b; Peng et al., 2024b). Efficient extrapolation methods can be categorized into two types, those focusing on partial contexts and those training on much shorter contexts.

Focusing on Partial Contexts LongLoRA (Chen et al., 2024i) employs S^2-Attn with shift and grouping operations for local sparse attention while using LoRA for long-context scenarios. Zebra (Song et al., 2023) introduces local attention with global approximation, combining local attention windows with a global approximation for improved efficiency. LandmarkAttn (Mohtashami & Jaggi, 2023) innovatively uses landmark tokens as processing block gates, enabling inference at any context length. CREAM (Wu et al., 2024f) alleviates the "middle loss" problem in long context processing through middle sampling optimization. LongRecipe (Hu et al., 2024j) extracts shorter but information-dense segments by identifying tokens with significant impact in long context processing. FoT (Tworkowski et al., 2024) extends model context length by adding memory attention mechanisms to certain transformer layers and using kNN algorithms for key-value pair retrieval.

Training on Much Shorter Contexts GrowLength (Jin et al., 2023) applies progressive length growth during training, starting with shorter sequences and gradually increasing context length to improve training efficiency while achieving extrapolation. E^2-LLM (Liu et al., 2024i) supports longer context windows during inference by using position index scaling and offset while only requiring training on shorter sequences. FocusLLM (Li et al., 2024q) proposes a parallel decoding approach, reducing complexity to $1/n$ of the original by freezing initial parameters and adding minimal training parameters, improving length extrapolation capability through training on short context. PoSE (Zhu et al., 2023), RandPos (Ruoss et al., 2023), and CD-Pos (Hu et al., 2024i) enhance model capability in processing varied input lengths by extracting smaller segments and adjusting position embeddings within these windows during training.

2.4 Extrapolation without RoPE

2.4.1 NoPE-based Extrapolation

Research on NoPE has revealed that causal masking injects sequential constraints into the network, since each token only attends to preceding content which implicitly encodes positional information (Haviv et al., 2022; Chi et al., 2023). This observation motivates NoPE-based LLM. Experiments demonstrate that NoPE-based LLMs achieve comparable performance to traditional position embeddings in certain tasks (Kazemnejad et al., 2024).

However, NoPE also struggles with length extrapolation (Kazemnejad et al., 2024; Wang et al., 2024g). Research shows that when context length exceeds the training range, NoPE's attention distribution becomes dispersed, leading to performance degradation. To address this issue, Wang et al. (2024g) proposes an optimization method based on attention temperature parameters and improves length generalization capability.

2.4.2 Other works

NoPE challenges whether position embedding is necessary for length extrapolation. Besides, there are other discussions regarding position embedding or length extrapolation.

Other Position Embedding Schema Several studies have proposed novel position embedding schema to address length extrapolation challenges or model long context better. KERPLE (Chi et al., 2022) introduces a kernel-based relative position embedding. FIRE (Li et al., 2024h) improves the Transformer's generalization capability in longer contexts through progressive interpolation. DAPE (data-adaptive position embedding) and DAPE V2 (Zheng et al., 2024a;b) dynamically adjusts positional offset matrices based on input data. CoPE (Golovneva et al., 2024) allows positions to depend on context by computing attention through selectively incrementing positions incrementing positions. BiPE (Dong et al., 2024e) combine intra-segment and inter-segment embeddings, using the former to identify positions within segments and the latter to model relationships between segments. Similarly, HiRoPE (Zhang et al., 2024g) tries a hierarchical RoPE in long code.

Attention Entropy Researchers observe that attention entropy increases with context length (Han et al., 2024; Peng et al., 2024b), prompting several innovative solutions. Many researchers introduce scaling factors in attention logits to reduce attention entropy. ReRoPE (Su, 2023b) incorporates a dynamic scale factor $\log_T t$ (where T is the pre-training sequence length and t is the input token's position index) in attention logits. YaRN (Peng et al., 2024b) introduces a scale factor in attention logits. Entropy-ABF (Zhang et al., 2024s) employs a special treatment of scaling factors for the first two attention layers, based on the discovery that the first two attention layers consistently exhibited almost identical attention patterns, with only subsequent layers showing trends of attention concentration.

Beyond these two directions, Dong et al. (2024e) proposes two training-free methods, positional vector replacement, and attention window extension, to effectively extend context length. From a memory perspective, RMT (Bulatov et al., 2023) also extends input context length by adding memory tokens and segment-level recursion to pre-trained LLMs.

3 KV Cache Optimization

Although length extrapolation can theoretically extend the context length of LLMs, it is only the tip of the iceberg of long-context LLMs. In Transformer-based LLMs, the KV cache expands with the increase of context length, resulting in a great computational and memory overhead (Fu, 2024; Luohe et al., 2024; Xiao et al., 2024c). Since the size of the KV cache is determined by the product of ***cached sequence length***, ***number of layers*** (§3.3), ***number of KV heads*** (§3.4), ***number of feature dimensions*** (§3.5), and ***storage data type*** (§3.6) (Fu, 2024; Raman, 2024), we can optimize the overhead through each of these factors as shown in Figure 5. Particularly, since the optimizations over sequence length are most discussed, we divide them into ***toke dropping*** (§3.1) and ***token merging*** (§3.2).

3.1 Token Dropping

Token dropping is a technique that identifies *unimportant* tokens and discards them. However, the critical challenge in these methods lies in determining which tokens are *unimportant*. Generally, token classification strategies can be categorized into two main types: static (Xiao et al., 2024c; Han et al., 2024) and dynamic (Zhang et al., 2023f; Li et al., 2024n).

For static strategies, token importance is considered independent of context, with certain tokens at specific positions consistently receiving more attention from the LLMs, and thus

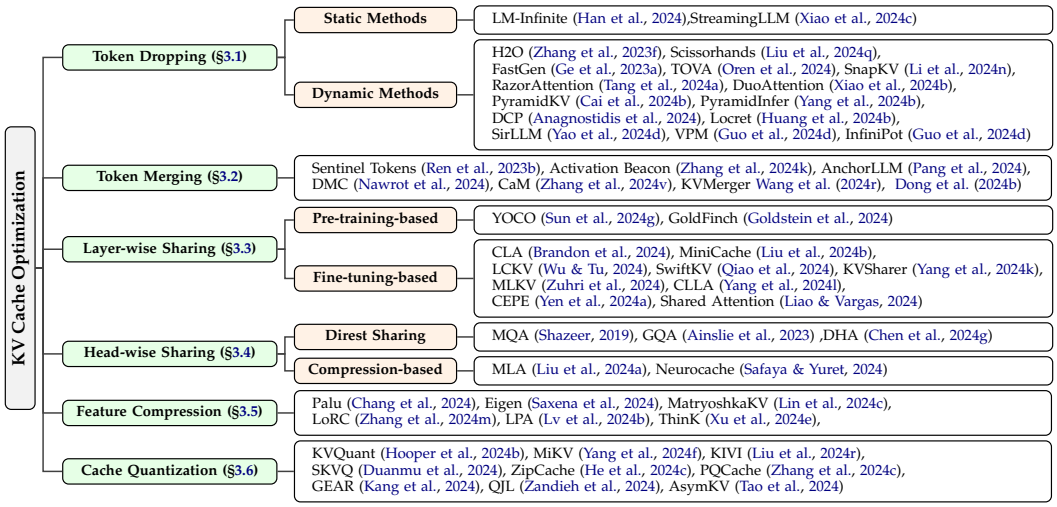

Figure 5: An overview of KV cache optimization of long-context LLMs.

being deemed *important*. For instance, sliding window attention (Jiang et al., 2023a; Bai et al., 2023a) retains the most recent tokens. Building on this, StreamingLLM (Xiao et al., 2024c) and LMInfinite (Han et al., 2024) observe that the initial tokens also consistently attract more attention from the LLM. Retaining both the most recent and the initial tokens helps mitigate the degradation of LLM performance as context length increases.

In contrast, dynamic strategies adaptively select *important* tokens based on their context. A commonly employed approach involves determining token importance using attention weights. For instance, H2O (Zhang et al., 2023f) identifies important tokens through cumulative normalized attention scores while prioritizing the retention of the most recent tokens. Scissorhands (Liu et al., 2024q) identifies pivot tokens via attention weights, ensuring that the memory usage of the KV cache remains within a fixed budget.

Due to the inherent flexibility of dynamic approaches, the majority of subsequent work has built upon and extended these methods. In addition to using attention weight as a measure of importance, researchers have identified variations in attention patterns across different attention heads and developed more refined criteria for determining token importance. For example, FastGen (Ge et al., 2023a) classifies attention heads into five types and applies distinct token eviction strategies for each. TOVA (Oren et al., 2024) removes tokens with the lowest attention scores for each head independently. SnapKV (Li et al., 2024n) selects queries within a local window and votes on the importance of previous tokens for each query and head. RazorAttention (Tang et al., 2024a) and DuoAttention (Xiao et al., 2024b) categorize attention heads into retrieval and non-retrieval heads, prioritizing the retention of initial and recent tokens for non-retrieval heads. Rehg (2024) takes this a step further, introducing different eviction rates for different attention heads.

Other researchers have considered the variability of attention patterns across layers and made corresponding adjustments. PyramidKV (Cai et al., 2024b) retains more tokens in lower layers, creating a pyramid-like KV cache structure, while PyramidInfer (Yang et al., 2024b) extends this by applying token-dropping strategies in deeper layers. SimLayerKV (Zhang et al., 2024r) focuses on identifying which layers can adopt the StreamingLLM (Xiao et al., 2024c) paradigm and drops intermediate tokens in the corresponding layers. In recent work, SCOPE (Wu et al., 2024c) optimizes KV cache usage separately for the pre-filling and decoding stages.

Beyond attention weights as a measure of token importance, researchers have also explored alternative metrics that may better capture this concept. DCP (Anagnostidis et al., 2024) fine-tunes a low-dimensional QK mapping to determine which tokens to drop. Similarly, Locret (Huang et al., 2024b) fine-tunes a new retention head to prioritize token retention. SirLLM (Yao et al., 2024d) utilizes token entropy to decide whether to discard a token.

Devoto et al. (2024) employs the L2 norm of keys to assess token importance. RoCo (Ren & Zhu, 2024) uses the standard deviation of attention scores as a metric for importance. VPM (Guo et al., 2024d) considers not only attention weights but also the values themselves. InfiniPot (Guo et al., 2024d) evaluates token importance based on a combination of future confidence and overlap with past information.

3.2 Token Merging

The methods discussed here focus on preserving the information of discarded tokens as much as possible through token merging, which can be seen as an extension of the token-dropping strategies mentioned earlier.

Sentinel Tokens (Ren et al., 2023b) introduces sentinel tokens to compress contextual information within segments. Similarly, approaches like Activation Beacon (Zhang et al., 2024k) and AnchorLLM (Pang et al., 2024) adopt analogous strategies, introducing special tokens to guide LLMs in learning how to effectively compress the KV cache during training, thereby achieving impressive performance. Dong et al. (2024b) uses kernel functions to compress preceding contextual information. DMC (Nawrot et al., 2024) fine-tunes decision and weight variables to determine when to expand the KV cache or aggregate weights into the final set of KV caches. Wang et al. (2024r) observes the similarity between adjacent keys and employs Gaussian kernel functions to merge neighboring tokens.

3.3 Layer-wise Sharing

For optimizations targeting the layer dimension, some approaches involve pre-training LLMs from scratch, while others focus on fine-tuning pre-trained models.

Sharing the KV cache across multiple layers is a common strategy for methods that modify the model architecture during the pre-training stage. YOCO (Sun et al., 2024g) divides the decoder into self-decoder and cross-decoder layers. KV cache is generated only in the output layer of the self-decoder, while cross-decoder layers reuse the output from the final self-decoder layer, thereby eliminating the need for additional KV caches. Similarly, GoldFinch (Goldstein et al., 2024) adopts a related strategy, where the last one-third of the layers utilize a small, compressed global KV cache generated by preceding layers. CEPE (Yen et al., 2024a) stores the full KV cache for the main input across all layers, while for additional context, each layer shares a small encoder output cache to perform cross-attention.

For fine-tuning existing LLMs, researchers often adopt straightforward inter-layer cache-sharing strategies. CLA (Brandon et al., 2024) uses fine-tuning to enable multiple layers to share the KV cache of a single layer. Additionally, methods such as MiniCache (Liu et al., 2024b), LCKV (Wu & Tu, 2024), KVSharer (Yang et al., 2024k), and SwiftKV (Qiao et al., 2024) adaptively select inter-layer cache sharing strategies. MLKV (Zuhri et al., 2024) combines layer-wise KV sharing with MQA, integrating adjacent layer sharing with techniques that replace deep-layer KV with shallow-layer KV. CLLA (Yang et al., 2024l) extends MLA and CLA by incorporating quantization into the shared caching mechanism. In contrast, CEPE (Yen et al., 2024a) employs a distinct strategy, storing a single-layer KV cache for all layers by encoding the KV cache with the representation generated by an encoder and integrating it with cross-attention.

Beyond KV cache sharing, researchers have also explored alternative strategies. Shared Attention (Liao & Vargas, 2024) directly shares attention weights across different layers to optimize performance along the layer dimension.

3.4 Head-wise Sharing

Similar to layer dimension optimizations, reducing the number of heads significantly impacts the representational capacity of LLMs. To preserve performance, head dimension optimizations typically rely on sharing strategies. For instance, GQA (Ainslie et al., 2023) and MQA (Shazeer, 2019) reduce memory usage by sharing the KV cache across queries from different heads, a technique now widely adopted in various model architectures. Ad-

ditionally, fine-tuning existing models can further optimize the size of the head dimension. For example, SHA (Cao et al., 2024) computes the cosine similarity of head weight matrices and groups similar heads to share a single KV cache. DHA (Chen et al., 2024g) employs a centroid alignment method to compute head similarity, linearly fusing the KV caches of similar heads, effectively compressing MHA into GQA.

Beyond KV cache sharing, low-rank compression is frequently used to optimize the head dimension. MLA (Liu et al., 2024a) replaces the full KV cache with low-dimensional latent vectors, recovering the KV through a projection matrix and injecting positional information via decoupled RoPE. ECH (Yu et al., 2024a) applies SVD-based low-rank decomposition to grouped head weight matrices, achieving a KV compression effect similar to GQA, but distinct in its non-averaging fusion. Neurocache (Safaya & Yuret, 2024) applies low-rank compression to head matrices and uses the most similar caches in attention computation.

3.5 Feature Compression

Optimization methods targeting feature dimensions primarily focus on low-rank compression, which corresponds to the size per attention head. Palu (Chang et al., 2024) introduces a medium-grained grouped head low-rank decomposition (G-LRD) method, striking a balance between accuracy and reconstruction efficiency. Eigen Attention (Saxena et al., 2024) utilizes a small calibration dataset to select the most significant directions based on SVD. MatryoshkaKV (Lin et al., 2024c) addressed the limitations of PCA by fine-tuning the orthogonal projection matrix to align the model outputs as closely as possible with the original outputs. Additionally, it employed a Matryoshka hierarchical strategy to achieve improved compression without sacrificing performance. LoRC (Zhang et al., 2024m) similarly leveraged SVD, adjusting cumulative condition numbers layer by layer to evaluate and modify compression ratios from deep to shallow layers, effectively preventing error accumulation that could degrade overall performance. In contrast, LPA (Lv et al., 2024b) focused on incorporating low-rank projection attention structures during pretraining, thereby improving performance on downstream tasks. ThinK (Xu et al., 2024e) introduces a dimension-pruning approach for feature compression, evaluating the interaction strength between KV pairs to retain the most significant dimensions.

3.6 Cache Quantization

Quantization is one of the most widely used techniques for KV cache compression, commonly adopted in practice for its speed and efficiency (Bai et al., 2023a; GLM et al., 2024). This optimization focuses on adjusting the size of the KV cache data type, which directly influences the storage size per unit.

Some works adapt traditional quantization methods to the specific characteristics of the KV cache. For example, KVQuant (Hooper et al., 2024b) determines quantization parameters through offline data analysis, ensuring that critical information is preserved during the process. In contrast, KIVI (Liu et al., 2024r) exploits the differing characteristics of keys and values in the model, performing channel-wise quantization for key caches and token-wise quantization for value caches. MiKV (Yang et al., 2024f) combines eviction strategies by storing tokens scheduled for eviction at a lower precision. SKVQ (Duanmu et al., 2024) rearranges key-value pairs to group outliers together, then trims boundary values within these groups to minimize quantization errors. ZipCache (He et al., 2024c) improves the compression ratio by normalizing attention scores within a channel-separable quantization framework. PQCache (Zhang et al., 2024c) integrates embedding retrieval techniques by decomposing the original vector space into Cartesian products of several lower-dimensional vector spaces, which are quantized separately.

Other approaches explore more advanced possibilities in quantization methods. For instance, GEAR (Kang et al., 2024) further reduces errors compared to full-precision computations by using low-rank and sparse matrices to fit residuals on top of traditional quantization results. QJL (Zandieh et al., 2024) introduces a novel KV cache quantization technique optimized specifically for CUDA kernels, enhancing the quantization process's efficiency and making it more suitable for large-scale parallel computing environments. AsymKV (Tao

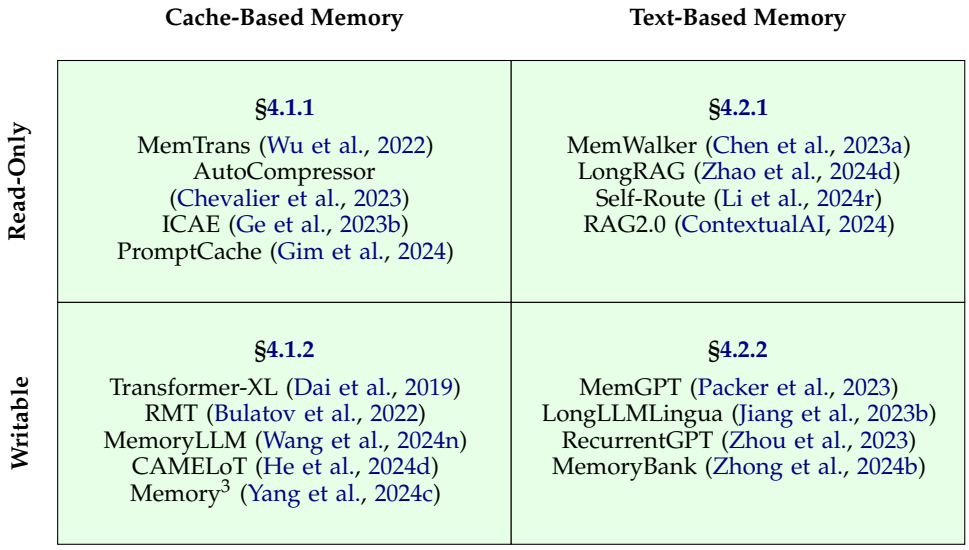

Figure 6: An overview of memory management of long-context LLMs.

et al., 2024) proposes an asymmetric quantization strategy that enables KV cache operation with extremely low 1-bit precision.

4 Memory Management

While KV cache optimization strives for a longer context practically, essentially, it is a balance between efficiency and performance. Cache optimization does not try to break the ceiling of LLM capabilities, since it does not change the organizing form of contextual information (Fu, 2024; Luohe et al., 2024). Long-context LLMs based on vanilla KV cache mechanism still face limitations including read-only access and the requirement to read all information at once, making them unsuitable for more complex scenarios (Dai et al., 2019; Bulatov et al., 2022). This has led to incorporating ***memory management*** into LLMs, with the KV cache being regarded as a specific memory instance.

Memory management in LLMs can be categorized from two perspectives. One is ***cache-based memory*** (§4.1), storing intermediate results that encode contextual information, such as KV cache, or ***text-based memory*** (§4.2), storing text directly, which is more convenient and flexible, as it allows the use of external textual data sources. The other is ***read-only*** or ***writable***, based on whether the memory is modifiable during storage. These two aspects divide the memory management methods into four quadrants as shown in Figure 6.

4.1 Cache-Based Memory

In this subsection, memory primarily refers to intermediate computational outputs, including hidden states, KV cache, and compressed textual representations that are irrecoverable.

4.1.1 Read-Only

The most intuitive improvement of read-only memory over the KV cache is its more flexible access method, avoiding reading all KV cache at once. MemTrans (Wu et al., 2022) stores the KV cache of pre-training in external memory to provide more relevant information during inference. MemLong (Liu et al., 2024m) extends this concept to a long context by storing the KV cache of context chunks and retrieving KV pairs based on relevance to guide inference.

Another approach to applying memory to long contexts is to compress the context, ensuring that the LLMs can handle longer sequences. AutoCompressor (Chevalier et al., 2023)

iteratively processes the context by encoding each segment into a fixed-dimension summary vector and concatenating it with the next part. Later works, such as LLoCO (Tan et al., 2024b) and E2LLM (Liao et al., 2024b), extend this method with advancements in offline learning and parallel compression, respectively. ICAE (Ge et al., 2023b) compresses information by fine-tuning the encoder to encode the entire context into a small number of memory tokens. UIO-LLMs (Li et al., 2024k) further conceptualizes memory-enhanced LLMs as fully connected RNNs, optimized through backpropagation.

Additionally, some inference acceleration works have also used memory. PagedAttention (Kwon et al., 2023) accelerates inference by reusing the same prefix of KV cache in a single request. Prompt Cache (Gim et al., 2024) and SGLang (Zheng et al., 2024c) speed up inference through structured organization of prompts to enhance performance.

4.1.2 Writable

In contrast to read-only memory, writable memory allows dynamic adjustments to stored memories. Transformer-XL (Dai et al., 2019), for example, reuses the hidden states of previous segments to capture long-term dependencies. RMT (Bulatov et al., 2022) improves upon this by introducing special memory tokens to store contextual information, with cross-segment gradient backpropagation to update the memory. Bulatov et al. (2023) extends the context length to 1M tokens using RMT. UniMem (Fang et al., 2024a) further synthesizes previous methods, while MemoryLLM (Wang et al., 2024n) and CAMELoT (He et al., 2024d) optimize memory management through more flexible or non-training-based approaches.

As researchers focus on using memory to store contextual or long-term information, Memory^3 (Yang et al., 2024c) was the first to introduce knowledge to LLMs and decompose knowledge into abstract knowledge and specific knowledge, formalizing the idea that the LLMs can store only abstract knowledge, while all specific knowledge is stored externally. This external memory is accessed during inference by periodic concatenation of relevant memories, achieving state-of-the-art performance.

4.2 Text-Based Memory

While cache-based memory has proven effective, it is relatively complex and lacks sufficient interpretability, particularly due to its non-textual nature. Thus, some researchers have turned to text-based memory to enhance LLMs' performance.

4.2.1 Read-Only

A common application of text-based memory is the presence of ground truth in text, where providing this text to the LLMs during generation can improve performance. Retrieve Augmented Generation(RAG, (Lewis et al., 2020)) utilizes this idea by retrieving external information using a retriever and appending it to the prompt during generation, paving the way for subsequent developments. This idea has been expanded to address long-context problems by retrieving relevant context segments (Chen et al., 2023a), improving queries (Fei et al., 2024), combining query and context (Zhao et al., 2024d), and improving retrieval methods (Luo et al., 2024; Soh et al., 2024; Jiang et al., 2024c), effectively addressing long-context challenges.

While RAG-related research has flourished, some studies have questioned the necessity of using RAG. Li et al. (2024r) conducted experiments revealing that performance with long-context LLMs outperforms RAG, suggesting an LLM-driven decision of whether to reuse long-context responses after initially employing RAG. The question of whether long-context or RAG is better remains a topic of ongoing discussion, which will be addressed later in **Q4** in Section 12. Some argue that RAG is more suitable for resource-constrained scenarios compared to long-context, and we will also present our perspectives on this matter in Section7. ContextualAI (2024) integrates various RAG components and conducts end-to-end training, achieving state-of-the-art results.

4.2.2 Writable

Writable text-based memory can be used to store and update historical information. MemoryBank (Zhong et al., 2024b) stores user history and profiles, achieving better user preference. Inspired by LSTM, RecurrentGPT (Zhou et al., 2023) summarizes preceding content during each step, facilitating ultra-long text generation. MemGPT (Packer et al., 2023) designs a multi-layered memory architecture, structuring prompts based on operating system memory access principles. EM^2 (Yin et al., 2024c) was the first to recognize that the direction of memory updates is not always optimal, introducing the EM algorithm (Dempster et al., 1977) and treating memory as latent variables to estimate the correct update direction.

Some researchers have also used memory to compress long contexts. One approach, which we refer to as text-level compression, involves compressing the context into several complete texts. Researchers have explored content-based compression (Fei et al., 2023), relevance-based compression (Yoon et al., 2024), and attention-weighted compression (Choi et al., 2024), achieving promising results. Another approach, token-level compression, compresses context into tokens that may not form complete sentences. LongLLMLingua (Jiang et al., 2023b) and Perception Compressor (Tang et al., 2024b) select the most relevant content based on correlations, retaining only the most important tokens to achieve token-level compression. Selection-p (Chung et al., 2024) retains a proportion of the original context tokens and trains the LLMs to generate responses using this limited set of tokens, resulting in significant improvements.

5 Architecture Innovation

Although KV cache optimization (Section 3) and memory management (Section 4) have improved the long-context capability of Transformer-based LLMs. The inherent shortage of Transformer in computation and memory efficiency still drives researchers to explore innovations in the attention mechanism itself, resulting in more radical architecture innovations (Jiang et al., 2024a; Ye et al., 2024a; Peng et al., 2023a; Gu & Dao, 2023). In this section, we will demonstrate those architectural innovations concerning long-context efficiency or performance from three perspectives as shown in Figure 7.

- In §5.1, we will analyze ***efficient attention***, the attention variant towards better computational efficiency or long-context performance. It can be further divided into two branches. One is ***attention approximation***, an efficient approximation for standard attention, such as MInference (Jiang et al., 2024a), RetrievalAttention (Liu et al., 2024d) and other sparse attention methods (Yang et al., 2024i; Zhu et al., 2024), while the other is ***attention alternative***, which tries a novel attention mechanism like DIFF-Transformer (Ye et al., 2024a), Lightning Attention (Qin et al., 2024d;c) and other linear attentions (Katharopoulos et al., 2020).
- As a cache-free architecture, discussion on LSTM (Schmidhuber et al., 1997) is revived for the pursuit of long context. In §5.2, we will analyze researches on LSTM in the LLM era, including the ***module-level Improvements*** like xLSTM (Beck et al., 2024) and HGRN series (Qin et al., 2024f;e) and the ***model-level advancements***, namely RWKV series (Peng et al., 2023a; 2024a; Choe et al., 2024).
- In §5.3, we will show the developing path of the widely-discussed Mamba series (Gu & Dao, 2023; Dao & Gu, 2024; Wang et al., 2024h), from the ***theoretical basis*** such as HiPPO (Gu et al., 2020) and S4 (Gu et al., 2021a) to its improvements (Ben-Kish et al., 2024; Yuan et al., 2024a), then to the ***hybrid architectures*** (Dong et al., 2024d; Akhauri et al., 2024), including Jamba series (Team et al., 2024c; Lieber et al., 2024)

5.1 Efficient Attention

5.1.1 Attention Approximation

Attention approximation is a hot research topic in long-context LLMs. Most attention approximation approaches are achieved with dynamic sparse attention through retrieval-

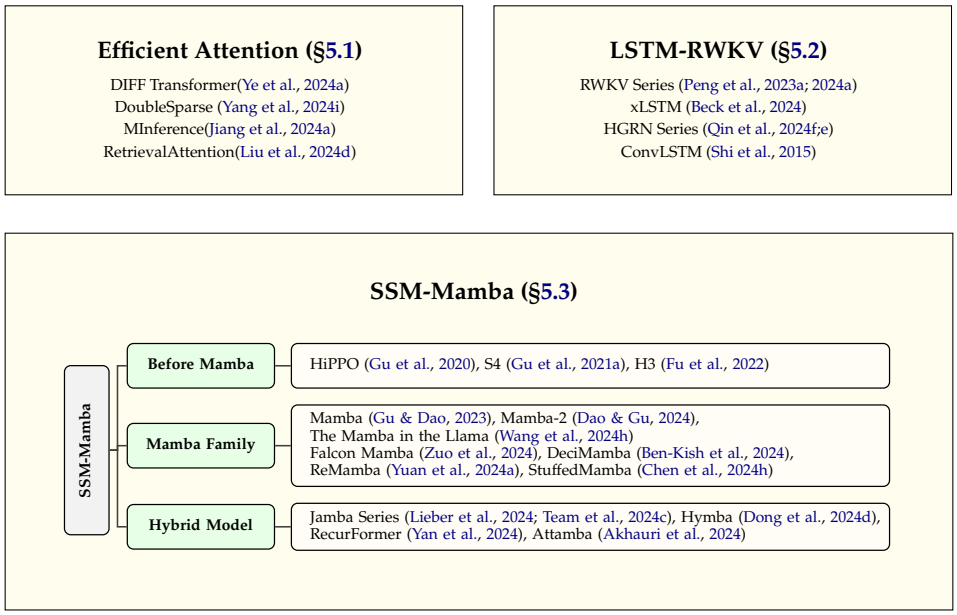

Figure 7: An overview of architecture innovation in long-context LLM.

based (Ribar et al., 2023; Liu et al., 2024d) or attention pattern observation (Jiang et al., 2024a). For example, SparQ Attention (Ribar et al., 2023) optimizes the attention mechanism through approximate attention computation based on KV cache extraction and interpolation compensation. Similarly, Loki (Singhania et al., 2024) ranks and selects tokens in the KV-cache based on attention scores computed in low dimensional space. Moreover, SampleAttention (Zhu et al., 2024) proposes a two-stage sampling filtering mechanism, identifying important attention patterns through query sampling, then combining selected KV cache with sliding windows. DoubleSparse (Yang et al., 2024i) uses important feature channels to identify key tokens, thereby reducing access to the KV cache. RetrievalAttention (Liu et al., 2024d) identifies the inconsistency between query and key vector distribution and resolves it through approximate nearest neighbor search. MagicPIG (Chen et al., 2024l) utilizing locality-sensitive hashing (LSH) sampling to estimate attention layer outputs. SqueezedAttention (Hooper et al., 2024a) optimizes attention computation by identifying the most important keys through semantic clustering and hierarchical lookup.

Other attention approximation approaches are achieved with dynamic sparse attention based on further observation of attention pattern (Jiang et al., 2024a). For example, MInference (Jiang et al., 2024a) proposes dynamic sparse attention from the perspective of sparse patterns. StarAttention (Acharya et al., 2024) proposes dividing the input into chunks distributed across different hosts for local attention computation, followed by aggregating global attention results through designated query hosts. And some works improve attention computation efficiency by using full attention and sparse attention in different layers or different heads (Beltagy et al., 2020; Li & Chan, 2019; Ainslie et al., 2020). Additionally, Fourier Transformer (He et al., 2023) removes redundant contextual information from hidden states by discrete cosine transform (DCT) to reduce computational complexity.

5.1.2 Attention Alternative

In this part, we will present methods that modify the fundamental mathematics of dot-product attention as attention alternative mechanisms, which require LLM pre-training from scratch but offer theoretical guarantees of improved efficiency (Choromanski et al., 2020). A representative work is linear attention (Katharopoulos et al., 2020), which reformulates dot-product attention using kernel functions to achieve linear complexity. In the LLM era, several recent studies propose novel approaches. SLAB (Guo et al., 2024c) optimizes attention computation efficiency through simplifies linear attention and progressive Layer-

Norm replacements. Lightning Attention (Qin et al., 2024d) achieves efficient computation by blocking and using linear attention between blocks. Its improved version, Lightning Attention-2 (Qin et al., 2024c), achieves the ability to process infinite-length contexts by introducing an exponential decay mechanism in the KV cache. Gated Slot Attention (Zhang et al., 2024t) enhances ABC (Peng et al., 2022) by incorporating a gating mechanism, essentially comprising a two-layer GLA (Yang et al., 2024j) linked via softmax to achieve more efficient memory utilization. What's more, DIFF Transformer (Ye et al., 2024a) calculates attention scores as the difference between two separate softmax attention maps. This subtraction eliminates noise and promotes the emergence of sparse attention patterns.

Furthermore, a recent study (Yang et al., 2024g) reveals an important insight: the efficiency of efficient attention, both sparse and linear attention, is task-dependent, with advantages primarily manifesting in tasks exhibiting locality characteristics. This finding opens new perspectives for research in efficient attention mechanisms.

5.2 LSTM-RWKV

Despite numerous advances in Transformer's computational efficiency, significant storage limitations persist (Ribar et al., 2023; Yang et al., 2024i). This leads researchers to explore *cache-free* architectures, with improvements of LSTM (Graves & Graves, 2012) emerging as a key direction. Compared to the Transformer's quadratic complexity, LSTM's linear inference complexity demonstrates significant advantages in long context scenarios. The improvements encompass both module-level enhancements to the basic LSTM architecture and large-scale innovations exemplified by RWKV (Peng et al., 2023a; 2024a), which shows exceptional performance in complex reasoning tasks like Sudoku[2].

5.2.1 Module-level Improvements

For example, xLSTM (Beck et al., 2024) consists of two parts. sLSTM introduces exponential gating, normalization, and stabilization mechanisms while supporting multi-head processing, significantly enhancing LLM's expressiveness while maintaining parallelism. Meanwhile, mLSTM further expands the cell state from vector to matrix form, giving LLMs stronger memory capabilities. Based on the xLSTM architecture, xLSTM-Mixer (Kraus et al., 2024) further introduces normalization and initial linear prediction mechanisms, enhancing LLM's performance by combining original embeddings and reverse embeddings. HGRN (Qin et al., 2024f) emphasizes the importance of forget gates in recursive layers, achieving hierarchical modeling of long-short term dependencies through learnable, layer-increasing lower bound values. Furthermore, HGRN2 (Qin et al., 2024e) innovatively introduces an outer product-based state expansion mechanism, expanding the scale of recursive states without increasing parameters, and addresses increased computational complexity through multi-head variants. Additionally, Feng et al. (2024) simplifies LSTM to enable parallel computation, improving LLM's computational efficiency.

Beyond these works, ConvLSTM (Shi et al., 2015) is an important direction for improvement. ConvLSTM demonstrates the viability and advantages of incorporating convolutional structures into LSTM. By implementing convolutional structures in both input-to-state and state-to-state transitions, ConvLSTM successfully extends LSTM to handle spatiotemporal context data. This innovation provides crucial insights for subsequent improvements of LSTM improvements (Wang et al., 2022; 2018; 2019; Lin et al., 2020).

5.2.2 Model-level Advancements

RWKV series represents a new technical approach, striving to combine the advantages of RNN and Transformer. RWKV4 (Peng et al., 2023a) introduces token shift, similar to convolutional sliding window processing, and processes context information through the fusion of time dimension (time-mixing) and feature dimension (channel-mixing). Its innovative WKV operator achieves training phase parallelization and linear complexity

[2]https://zeeklog.com/rwkv-tong-guo-ji-wan-token-de-cot-jie-jue-ji-hu-100-de-shu-du-wen-ti-cai-yong-29m-can-shu-de-xiao-mo-xing–2/

during inference. Subsequently, RWKV's development reaches new heights with RWKV5 (Eagle) and RWKV6 (Finch) (Peng et al., 2024a). RWKV5 introduces multi-head mechanisms similar to Transformer's multi-head attention mechanism and optimizes token shift through linear interpolation. In time-mixing, it enhances LLM's expressiveness by introducing new trainable parameters. RWKV6 further innovates with significant improvements in both token shift and time-mixing, particularly incorporating LoRA's implementation approach and allowing each channel to mix token information rather than relying on fixed trainable parameters. These improvements enable the LLM to demonstrate superior performance and higher efficiency in processing long contexts.

5.3 SSM-Mamba

State Space Model (SSM) represent an innovative architecture that delivers several key advances (Gu & Dao, 2023). Its linear computational complexity significantly outperforms the quadratic complexity of Transformers. It eliminates memory requirement for attention matrices through fixed hidden state storage. Most importantly, SSM supports parallel training and linear generation, offering substantial practical advantages.

SSM originates from modern control system theory. It encodes context information by maintaining hidden states and using linear dynamical systems to describe state evolution:$x'(t) = Ax(t) + Bu(t)$,$y(t) = Cx(t) + Du(t)$, where $x(t)$ represents hidden state, $u(t)$ represents input, $y(t)$ represents output, and A,B,C,D are parameter matrices.

5.3.1 Pre-Mamba Works

Although HiPPO (Gu et al., 2020) is initially applied to RNNs, it lays crucial theoretical foundations for the development of Mamba. HiPPO utilizes polynomial approximation and specific probability measures (LegS probability measure) to construct a new matrix structure (HiPPO matrix), effectively modeling context data by encoding historical information into polynomial coefficients. Building on this, LSSL (Gu et al., 2021b) further reveals the connection between RNN, CNN and SSM, discovering that SSM could be represented in both recurrent and convolutional forms. More importantly, LSSL first attempts to use HiPPO Matrix to initialize SSM's parameters, achieving significant performance improvements on multiple tasks. Then, the introduction of S4 (Gu et al., 2021a) marks a major breakthrough in SSM's computational efficiency. This work represents HiPPO matrix in NPLR (Normal Plus Low-Rank) form and reduces SSM's computational overhead from both recurrent and convolutional perspectives through matrix theory derivations. The subsequent S4D (Gu et al., 2022) proposes a simplified version of S4, further improving computational efficiency while maintaining LLM's performance by restricting the state matrix to a completely diagonal form. Later, H3 (Fu et al., 2022) focuses on addressing SSM's shortcomings in language modeling tasks. Inspired by linear attention mechanisms, H3 represents the update of SSM's hidden state as $Q \odot SSM_{diag}(SSM_{shift}(K)) \odot V$, where the two SSM matrices employ the "hungry hippo" mechanism to enhance efficiency. H3's performance in synthetic language modeling tasks matches attention mechanisms. Additionally, H3 introduces FlashConv to extend context length and improve training efficiency. In the above architecture innovations based on recurrent networks, local information interaction such as token shift often appears. Based on this common property, we will further the discussion on new architecture in **Q5** in Section12

5.3.2 Introduction and Improvements of Mamba

The introduction of Mamba (Gu & Dao, 2023) represents a significant milestone in SSM's development. It introduces a selective mechanism and enables content-aware capabilities. Specifically, when updating parameter matrices, Mamba incorporates projection information of inputs, allowing each token to have independent parameter matrices. Simultaneously, Mamba proposes a hardware-aware parallel recursive algorithm to improve computational efficiency. Mamba-2 (Dao & Gu, 2024) further improve the architecture, elucidating the dual relationship between Mamba and attention mechanisms through detailed theoretical analysis and providing insights for the integrated use of attention mechanisms and Mamba.

However, as research deepened, researchers discover Mamba's limitations in processing long contexts. Several works propose solutions from different angles. DeciMamba (Ben-Kish et al., 2024) proposes a token selection mechanism based on Δ_t by analyzing Mamba's receptive field. ReMamba (Yuan et al., 2024a), inspired by KV cache's compression method, uses architecture's characteristic of aggregating information through hidden states to select the most representative representations using importance score mechanisms. Stuffed-Mamba (Chen et al., 2024h) reveals the essence of the state collapse phenomenon, proposing multiple mitigation strategies including increasing state decay amount, reducing input information quantity, normalizing states, and simulating sliding window mechanisms.

Furthermore, researchers are exploring other optimization directions. SMR (Qi et al., 2024a) analyzes SSM's sampling stability issue from a control theory perspective, proposing an event-triggered control (ETC) based solution—introducing learnable memory to adjust current states and resolving Mamba's inability to use convolution, enabling efficient parallel computation. Mamba-PTQ (Pierro & Abreu, 2024) discovers the outlier channels problem in Mamba's quantization and uses SmoothQuant technology, balancing weight and activation quantization difficulty through transfer factor α. Additionally, The Mamba in the Llama (Wang et al., 2024h) uses the standard attention parameters to initialize Mamba, combining knowledge distillation and multi-step speculative decoding to improve efficiency.

5.3.3 Hybrid Architectures

Recently, researchers have explored hybrid architectures that combine SSM and Transformer. Early Jamba (Lieber et al., 2024) adopts a relatively direct approach, stacking Transformer, Mamba, and MoE blocks in combination, aiming to balance memory usage, computational throughput, and LLM's performance. RecurFormer (Yan et al., 2024) then proposes a more targeted hybrid solution, with its core idea being to identify and replace attention heads in Transformer that focus on local perception with Mamba blocks. Subsequently, Hymba (Dong et al., 2024d) proposes a deeper integration approach, adopting parallel Attention heads and SSM heads structure to avoid potential information bottleneck issues that might arise from serial architecture. And it achieves an organic fusion of the two types of heads through learnable parameters. Additionally, Attamba (Akhauri et al., 2024) explores a new compression approach that uses SSM blocks to compress multiple tokens into one chunk token for Transformer processing. And it also combines sliding window concepts to preserve the initial state of local tokens, thereby reducing KV cache.

Other New Architectures Beyond the aforementioned work, researchers also propose many other *cache-free* architectures, providing new perspectives for improving LLM's ability to process long contexts. Some works are based on Neural ODE (Chen et al., 2018a), such as Liquid Time-constant Networks (Hasani et al., 2021) introducing a dynamic adjustable liquid time constant mechanism and CfC (Hasani et al., 2022) avoiding the need for numerical solutions by finding approximate closed-form solutions for LTC. Additionally, MixCon (Xu & Lin, 2024) proposes a hybrid architecture combining Transformer layers, Conba layers, and MoE and introducing mechanisms such as selective state spaces to enhance LLM's performance. MCSD (Yang et al., 2024d) captures local and global features through Slope and Decay components respectively, and adopts a dual-branch design to strengthen feature extraction and fusion.

6 Training Infrastructure

Although architectural innovation has achieved great progress, the mainstream long-context LLMs are still based on Transformer (Dubey et al., 2024; AI, 2024; DeepSeek-AI, 2024) or hybrid architectures (Team et al., 2024c; MiniMax et al., 2025). Therefore, we need to make long-context training and inference possible while accepting the inherent drawback of the self-attention mechanism. To further the journey of extending context length, we turn our focus to the practical training and inference of long-context LLMs to explore infrastructure improvement. Whether for long-context training discussed in Section 6 or inference infrastructure discussed in Section 7, the focus of research all involve: computation, storage, and distribution, namely parallelism as shown in Figure 8 and Figure 9.

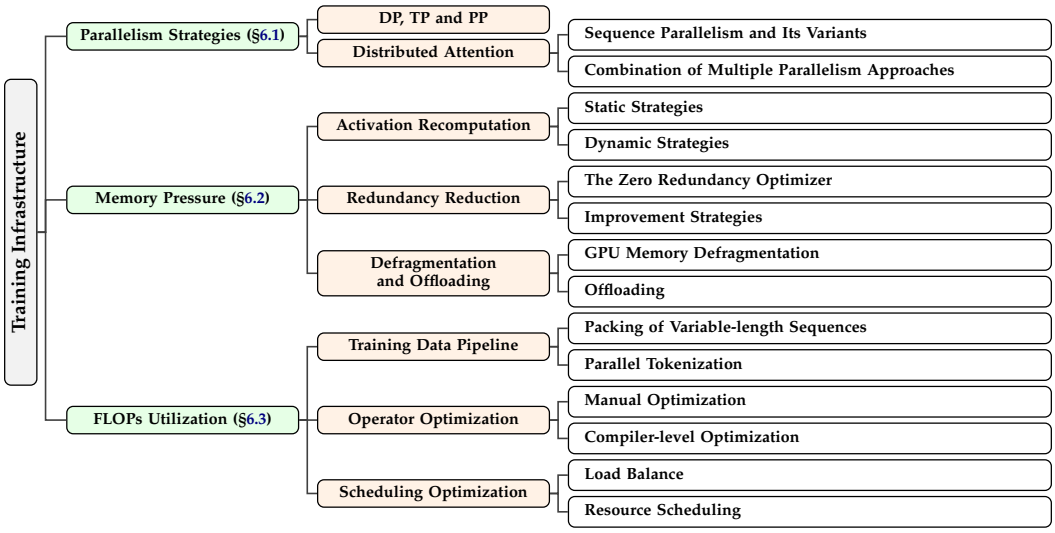

Figure 8: An overview of training infrastructure of long-context LLMs.

For example, for training infrastructure, currently, leading LLMs support context lengths exceeding 128k tokens (Meta, 2024a; Dubey et al., 2024; Yang et al., 2024a) and up to 256k tokens during pre-training (e.g., Qwen2.5 (Qwen et al., 2024)). At such a context length, the distributed parallelism strategies address the basic question of training possibility. Beyond that, handling such long sequences imposes significant memory demands and necessitates enhanced hardware utilization efficiency:

- GPU memory overhead scales proportionally with context length through activation values and optimizer states (Guo et al., 2024a; Duan et al., 2024). The demand for memory bandwidth intensifies due to larger tensor sizes (Patel et al., 2023; 2024a). The growth in GPU memory capacity and memory bandwidth has consistently fallen behind advances in GPU computational power (Gholami et al., 2024; Patel et al., 2023), further exacerbating the aforementioned challenges.
- Memory-Flops Utilization (MFU) represents the ratio of actual computational use to theoretical hardware performance. Large-scale long-context distributed training introduces considerable computational & communication overhead (Gu et al., 2024a; Sun et al., 2024a), reducing MFU. Accommodating longer contexts typically necessitates smaller batch sizes, thereby decreasing throughput.

We will briefly review mixed-precision training work (Narang et al., 2017; Kalamkar et al., 2019; Sun et al., 2019; Peng et al., 2023b; Dubey et al., 2024; DeepSeek-AI, 2024) at the end of this section, as it reduces GPU memory requirements and increases MFU, however expanding the supported context length of current training systems only indirectly.

6.1 Distributed Parallelism Strategies

Training modern AI models with extensive context windows has become increasingly complex, pushing beyond what single GPUs can handle (Patel & Nishball, 2024). This challenge has led to the development of sophisticated distributed training approaches, particularly when dealing with long context.

6.1.1 Data, Tensor, and Pipeline Parallelism

The foundational and most widely adopted approaches in the distributed parallelism are (Patel & Nishball, 2024): Data Parallelism (DP), which distributes input data across multiple GPUs (Li et al., 2020; Zhao et al., 2023c; Zhang et al., 2024n; Sun et al., 2024e); Tensor Parallelism (TP), which splits model parameters matrices across devices; and Pipeline Parallelism

(PP), which distributes model layers across GPUs. While each approach offers distinct advantages, they also present unique challenges. TP, for instance, effectively manages memory constraints but typically requires high-bandwidth communication between devices (Dong et al., 2024a). Similarly, PP often encounters efficiency losses due to pipeline bubble, and efforts are being made to eliminate this problem (Li et al., 2021b; Qi et al., 2024b; Arfeen et al., 2024). Sun et al. (2024a) schedules the pipeline of training LLMs at the sequence level on sequences up to 64k, reducing pipeline bubbles and memory footprint.

6.1.2 Distributed Attention

Sequence Parallelism (SP), specifically designed for long-context training, partitions input and output tensors along the sequence dimension at the Transformer layer level. It facilitates distributed processing of attention computations (Li et al., 2021a) and other operations (Shoeybi et al., 2019). Bian et al. (2021) introduced a sequence dimension partitioning and parallelization scheme. Ring Attention (Li et al., 2021a) then employs block-wise attention computation combined with a ring communication pattern to partition QKV tensors along the sequence dimension, distributing computation across devices. Ring Attention can be integrated with FlashAttention (Dao et al., 2022; Dao, 2024), preserving IO-awareness and memory efficiency. Ring attention with block-wise transformers (Liu et al., 2023a) further enhances the overlap between communication and computation, enabling the training of sequences exceeding 100 million tokens. To address Ring Attention's load imbalance in causal attention mask scenarios, several optimization (Brandon et al., 2023; Li et al., 2024c; Fang & Zhao, 2024; Gu et al., 2024a) solutions have emerged. Alternatively, Megatron-LM (Shoeybi et al., 2019) achieves load balancing through input token reordering.

Ulysses-Attention (Jacobs et al., 2023) introduces head-parallel stratification atop sequence dimension partitioning, enabling parallel attention head processing across GPU devices. The 2D-Attention mechanism (Gu et al., 2024a) resolves head-parallel strategy scalability limitations while addressing efficiency constraints present in previous context-parallel approaches such as Brandon et al. (2023) and Li et al. (2024c). Sun et al. (2024d) tailored to linear attention-based language models, scales sequence length up to 4096k.

In practical implementations, ultra-long context(eg. longer than 256k) (Qwen et al., 2024) training typically requires a strategic combination of multiple parallelism approaches. For example, common configurations integrate tensor and sequence parallelism within individual nodes while implementing data parallelism across machines. This hybrid parallelism methodology (Shoeybi et al., 2019; Narayanan et al., 2021; Jacobs et al., 2023; Chen et al., 2024e; Singh et al., 2024; Fujii et al., 2024; Dubey et al., 2024) enables effective scaling to larger computing clusters, substantially enhancing pre-training and fine-tuning efficiency. However, existing automatic parallelism tools require further optimization for the unique computation and communication patterns characteristic of ultra-long context scenarios.

6.2 Alleviating GPU Memory Pressure

GPU memory constraints have emerged as a critical bottleneck in model training as context windows expand. This pressure stems primarily from (Gholami et al., 2024; Guo et al., 2024a; Duan et al., 2024):

- Model parameters themselves
- activation values and optimizer states
- inter-device communications
- temporary space allocations and GPU memory fragmentation

While not specifically designed for long-context processing, current solutions offer valuable insights for training such models. We will provide a concise overview.

6.2.1 Activation Recomputation

GPU memory usage scales with sequence length. Activation recomputation (Chen et al., 2016; 2024d) trades computational power for memory space, addressing memory constraints while potentially improving the compute-to-memory ratio and helping resolve memory bottlenecks.

Selective checkpointing (Korthikanti et al., 2023; PyTorch, 2024) methods preserve outputs from critical layers, such as attention modules (Li et al., 2024c), while recomputing other intermediate results as needed. Selective-Checkpoint++ (Gu et al., 2024a) significantly reduces memory usage while maintaining performance by adding attention modules to a whitelist and preserving their softmax outputs.

In contrast to static strategies, dynamic recomputation approaches determine which activation values to discard and recompute at runtime. Kirisame et al. (2020) and Hu et al. (2022) employs heuristic methods for runtime tensor eviction and recomputation, while Zhao et al. (2024c) uses a token-wise activation recomputation and swapping mechanism with linear programming to optimize, like, activation value recomputation.

6.2.2 Redundancy Reduction

The Zero Redundancy Optimizer (ZeRO) introduces a progressive sharding scheme to minimize memory redundancy (Rajbhandari et al., 2020). ZeRO-1 distributes optimizer states across GPUs, ZeRO-2 extends this to gradients, and ZeRO-3 further shards model parameters, effectively dividing the total memory overhead by the parallel dimension. While this comprehensive sharding minimizes redundancy, it increases communication overhead. Numerous other works (Wu et al., 2023; Luo et al., 2023; Chen et al., 2024f) have tackled communication efficiency and mitigated communication costs. ZeRO++ (Wang et al., 2023a) redundantly stores an additional set of secondary parameters on each node, enhancing communication efficiency through parameter prefetching. MiCS (Zhang et al., 2022) and Fully Sharded Data Parallel (FSDP) (Zhao et al., 2023c) shard all model state components within subgroups and replicate them between subgroups to reduce communication scale.

6.2.3 GPU Memory Defragmentation & Offloading

Device memory limits affect manageable sequence length, requiring techniques like fragmentation elimination and offloading to expand capacity.

GPU memory defragmentation falls into two categories: tensor-based method (Kirisame et al., 2020; Hu et al., 2022; Shu et al., 2023; Zhao et al., 2024c; Zhang et al., 2024f) and Virtual Memory Management (VMM). For tensor-based approaches, ROAM (Shu et al., 2023) optimizes operator execution order and tensor allocation strategies using efficient tree-structured algorithms to identify optimal execution plans. MEMO (Zhao et al., 2024c) and Coop (Zhang et al., 2024f) also address memory fragmentation while reducing overall memory consumption. VMM-based solutions, such as GMLake (Guo et al., 2024b) and PyTorch Expandable Segments (PyTorch, 2024), utilize low-level CUDA driver APIs (Perry & Sakharnykh, 2024) to consolidate non-contiguous memory blocks into larger, contiguous segments through virtual memory address mapping.

Offloading technologies include CPU and SSD approaches. CPU offloading encompasses Static Offloading (Pudipeddi et al., 2020; Ren et al., 2021) and Dynamic Offloading (Sun et al., 2022a; Li et al., 2022). SSD Offloading solutions (Rajbhandari et al., 2021; Jang et al., 2024; Liao et al., 2024a) enable training of trillion-parameter models beyond CPU offloading capabilities. Recent advancements have proposed comprehensive solutions for managing high activation value occupancy and memory fragmentation during training. Zhao et al. (2024c) employs token-level decisions to determine which activation values to recompute and which to transfer to CPU memory, utilizing integer programming for memory allocation and space reuse by leveraging the uniform structure of Transformer layers. Ulysses-Offload (Yao et al., 2024b) achieves substantial GPU memory reductions through its novel Distributed Attention with Fetching and Offloading mechanism, and leverages a dedicated double buffer design to overlap almost all fetching with computation.

6.3 Enhancing Model FLOPs Utilization

Despite access to large-scale GPU clusters, LLaMA3.1 (Dubey et al., 2024) achieves a mere 38-41% Model FLOPs Utilization (MFU), suggesting substantial room for optimization. These inefficiencies (Duan et al., 2024) are exacerbated when handling longer context (e.g. longer than 32k).

- Data processing operations, including sequence packing and tokenization, encounter significant challenges with extended sequences.
- Longer sequence length results in quadratic growth in attention computation complexity. The memory bandwidth of current accelerator cards lags behind this computational surge, leading to longer processing times and reduced MFU.
- Different sequence lengths from 2k to 128k and above complicate load balancing and efficient scheduling.

6.3.1 Training Data Pipeline for Long-Context Models

Processing longer sequences introduces specific challenges in the training data pipeline, particularly in text sorting, packing, and tokenization. While research in this area remains limited, the training data pipeline for long-context training is a critical challenge that warrants further investigation, as discussed in **Q7** in Section12.

Training only on long data hurts models' long-context performance (Gao et al., 2024d). The conventional approach of batch-packing sequences of similar lengths introduces potential training biases through length uniformity, while random long & short-sequence packing results in GPU underutilization. To address this, GLM-Long (ChatGLM, 2024) organizes batches based on computational complexity, ensuring uniform computational complexity across packages and significantly reducing GPU idle periods. Furthermore, GLM-Long employs layer accumulation techniques to mitigate sorting-induced biases and utilizes loss reweighting strategies to handle imbalanced data volumes across packages.

Tokenization inherently allows for parallel processing along the sequence dimension. ParallelTokenizer (Cai et al., 2024c; OpenLMLab, 2024) leverages this by implementing parallel tokenization.

6.3.2 Operator Optimization

Optimizing operators primarily involves enhancing the Transformer's core computation—the attention mechanism. FlashAttention (Dao et al., 2022; Dao, 2024) represents a significant advancement in this domain by optimizing memory access patterns through block-wise computations, enabling efficient use of on-chip fast memory. This approach reduces latency without compromising attention accuracy and eliminates quadratic memory complexity, thereby supporting long-context training. FlashAttention-3 (Shah et al., 2024) further optimizes for H100 GPUs by fully utilizing hardware features such as asynchronous WGMMA instructions. Simultaneously, normalization, dropout and feed-forward network (FFN) computations have undergone engineering optimizations (Liu et al., 2023a; Ma et al., 2024a; Shoeybi et al., 2019), often through operator fusion. For instance, the JAX implementation of Ring Attention with Blockwise Transformers (Liu et al., 2023a) incorporates operator fusion for FFN, enhancing computational efficiency.

Compiler-level optimizations have also made significant strides, particularly with OpenAI Triton (Tillet et al., 2019) and other frameworks (Dong et al., 2024c; Spector et al., 2024). Triton offers a Python-based programming language and an MLIR-based (Lattner et al., 2020) compiler enriched with built-in optimizations, facilitating the development of high-performance operators through a user-friendly interface. Additionally, compiler-level operator fusion, which often requires comprehensive computation graph information (Chen et al., 2018b; Ansel et al., 2024; Wu et al., 2024e), automates optimization processes, thereby improving MFU.

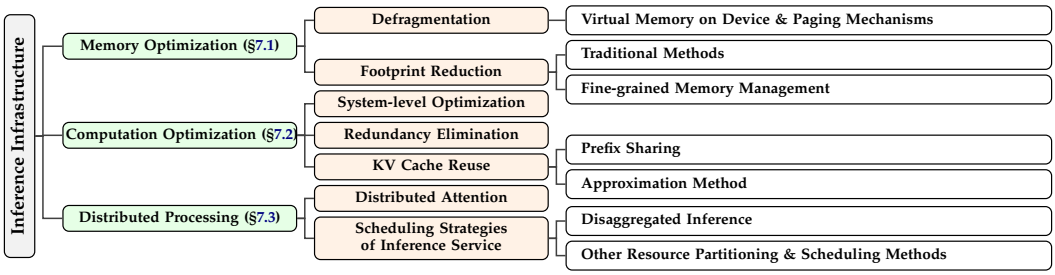

Figure 9: An overview of inference infrastructure of long-context LLMs.

6.3.3 Scheduling Optimization

Scheduling optimization is critical for enhancing training efficiency in long-context LLMs. As LLMs scale and context window size increases, factors such as computation-communication overlap (Wang et al., 2024d), load balancing, and CPU time significantly influence training speed (tokens per GPU per second) (Dubey et al., 2024; DeepSeek-AI, 2024). Given the limited research specifically for typical long-context, this section provides a concise overview.

Recent workload-scheduling developments have been tailored to LLMs. Xue et al. (2024a) optimizes concurrent training efficiency through hybrid parallel strategies and hardware affinity in heterogeneous clusters. Hydro (Hu et al., 2023) enhances hardware utilization through model scaling and consolidation, while Hu et al. (2024f) addresses mixed workload characteristics through solutions such as decoupled evaluation scheduling.

Resource-level improvements have also emerged. For example, SiloD (Zhao et al., 2023a) jointly allocates data caching and remote I/O as first-class resources, significantly improving system throughput.

mixed-precision training

In addition to the aforementioned methods, there are numerous approaches (Narang et al., 2017; Kalamkar et al., 2019; Sun et al., 2019; Dubey et al., 2024; Qwen et al., 2024; DeepSeek-AI, 2024) that improve the long context training throughput and MFU from the perspective of mixed-precision training. Wang et al. (2023b) explores 1-bit precision training. Recent hardware and framework developments (Xi et al., 2024; 2023; Jacobs et al., 2023; Shoeybi et al., 2019; Bian et al., 2021; Peng et al., 2023b; Liang et al., 2024c; Videau et al., 2024; NVIDIA, 2024a) have expanded support for lower precision operations (in FP8, FP4, INT4, etc.), offering new avenues for further enhancing MFU.

7 Inference Infrastructure

Developing effective strategies for long-context inference represents a strategic imperative for both industry and academia. In today's business landscape where API sales and Agent products dominate, efficient handling of longer contexts is essential (Barkley, 2024; Koh et al., 2024). Meanwhile, researchers have noted inherent limits on current pretrain paradigms (Dastin, 2024), especially as the growth of high-quality training data slows.

As context lengths extend to tens of thousands or even millions of tokens (Anthropic, 2024a; Reid et al., 2024), inference encounters bottlenecks including quadratic complexity of attention mechanism, KV cache storage demands, communication overhead and other challenges (Li et al., 2024a; Yuan et al., 2024c). These technical barriers directly impact inference systems' **throughput** and **latency**.

Researchers have tried to improve throughput by refining memory utilization (Sheng et al., 2023), optimizing batching techniques to maximize parallelism (Daniel et al., 2024), etc. At the same time, efforts to reduce latency include but are not limited to, minimizing

redundant attention calculations (Jiang et al., 2024a), reusing KV cache (Zheng et al., 2024c) and making the prefill and decode phases disaggregated (Jin et al., 2024b). Lastly, for contexts of hundreds of thousands or millions of tokens, there are scalable distributed solutions (Fang & Zhao, 2024; Lin et al., 2024b; Wu et al., 2024a).

This section ends with a curated overview of popular inference frameworks (Kwon et al., 2023; Contributors, 2023; Zheng et al., 2024c; Hugging Face, 2024; NVIDIA, 2024b) to guide readers in their research and deployment decisions, reflecting how today's inference engines have matured into sophisticated platforms that integrate recent findings (Yu et al., 2022; Dao, 2024; Dao et al., 2022; Agrawal et al., 2024; Jin et al., 2024b) with best engineering practices.

7.1 Memory Optimization

The pursuit of higher throughput has led us to optimize GPU memory usage in LLM inference systems (Patel & Nishball, 2024; Kwon et al., 2023; Zheng et al., 2024c), as the growing demands of processing long sequences pose some challenges for GPU memory, which we will discuss in the following.

7.1.1 GPU Memory Defragmentation

PagedAttention (Kwon et al., 2023) leverages virtual memory paging mechanisms similar to those operating systems use to manage the KV cache on fixed-size pages. TokenAttention (LightLLM, 2024; Hu et al., 2024d) manages the KV cache at the token level, achieving zero GPU memory waste. vAttention (Prabhu et al., 2024; Xu et al., 2024b), leverages CUDA's native virtual memory management capabilities (Perry & Sakharnykh, 2024), eliminates PagedAttention-like lookup tables, resulting in reduced latency.

7.1.2 Memory Footprint Reduction

Traditional Methods Chunk prefill (Agrawal et al., 2024; Holmes et al., 2024; Zeng et al., 2024b) divides long sequences into smaller blocks for gradual processing to reduce GPU memory pressure or batch them together with decoding requests to improve overall throughput. Approximate attention mechanisms, cache-free and other non-attention architectures shown in Section 5 can significantly reduce GPU memory costs for long-sequence computations and KV cache. Cache optimization techniques shown in Section 3 can substantially reduce deployment memory overhead while improving processing speed through low-precision advantages.

Fine-grained Memory Management Extended sequence length has necessitated more sophisticated memory management approaches. Researchers have introduced fine-grained memory management techniques (Sheng et al., 2023; He & Zhai, 2024; Jiang et al., 2024b; Gao et al., 2024a; Lee et al., 2024b). FlexGen (Sheng et al., 2023) uses linear programming to select optimal storage formats and access patterns for weights and attention cache.

The CPU memory and disk offloading (Liu et al., 2023b) need management too. Frameworks like DeepSpeed-inference (Aminabadi et al., 2022) and Huggingface Accelerate (Gugger et al., 2022) offload the weights of large models to CPU memory. Alizadeh et al. (2023) enables models up to twice the size of available DRAM to run by the combination of a low-rank predictor for selective neuron loading, a dynamic sliding window technique for caching activated neurons, and a row-column bundling mechanism to optimize data transfers between flash storage and DRAM.

7.2 Computation Optimization

Attention computation costs grow quadratically with sequence length, creating significant latency challenges for long-context inference (Beltagy et al., 2020; Liu et al., 2024o). Recent Studies address this by optimizing system-level implementation, reducing unnecessary calculations in attention and reusing existing results.

System-level Implementation Optimization this type of work focuses purely on engineering and implementation optimizations without modifying the underlying algorithms (Dao, 2024; Dao et al., 2022; Shah et al., 2024; Ye et al., 2024c; FlashInfer Community, 2024; Ye et al., 2025b; Gerganov et al., 2023). For example, FlashDecoding++ (Hong et al., 2023) accelerates flat GEMM (Wang, 2023; Ibrahim et al., 2024) with double buffering that overlaps computation and data transfer, hiding the memory latency in loading input matrices. Continuous batching (Yu et al., 2022; Daniel et al., 2024; Kwon et al., 2023) allows new sequences to be inserted into a batch whenever existing sequences complete their generation, yielding higher GPU utilization compared to static batching.

Computational Redundancy Elimination Research has revealed that attention patterns are notably sparse (Xiao et al., 2024c; Jiang et al., 2024a), with only a small subset of tokens significantly impacting next-token prediction. This insight has led to many optimization strategies that are discussed in Section 5.

KV Cache Reuse In practical applications, context often contains repetitive segments, while recalculating these increases latency with longer contexts (Gim et al., 2024). Early approaches used simple prefix matching for cache reuse or prefix sharing in decoding (Juravsky et al., 2024; Ye et al., 2024c), integrated into deployment frameworks (NVIDIA, 2024b; Ye et al., 2024b; FlashInfer Community, 2024; Lin et al., 2024d) rather than published as standalone work. RadixAttention (Zheng et al., 2024c) later improved this by organizing contexts in a radix tree structure, enabling efficient reuse with minimal CPU overhead and across requests. Another research direction employs approximation methods (Hu et al., 2024e; Yao et al., 2024a) to reuse KV cache across requests with partially matching prefixes, where identical segments are not contiguous. This requires careful handling of internal attention and position embedding while approximating cross-segment attention. For instance, EPIC (Hu et al., 2024e) introduced position-independent context caching, enabling flexible cache reuse across positions without affecting model accuracy.

7.3 Distributed Processing

When context length extends to hundreds of thousands or even millions of tokens (Yang et al., 2024a; Qwen et al., 2024; Reid et al., 2024; InternLM, 2024), the memory and computational capabilities of a single machine with a single GPU can no longer meet the demands. This section briefly discusses existing distributed solutions for enhancing long-context processing capabilities, focusing on Distributed Attention, scheduling strategies, and the increasingly popular Prefill-Decode (PD) disaggregation architecture.

7.3.1 Distributed Attention

Ring Attention (Li et al., 2021a) enables efficient processing of long sequences by splitting them across devices. Each device stores a portion of the KV cache, reducing GPU memory usage. Since the data transfer and computation can be fully overlapped through optimization (Liu et al., 2023a; Fang & Zhao, 2024), the additional communication overhead does not impact throughput. When combined with Context Parallel (Shoeybi et al., 2019), this method could enable a longer context.

Yang et al. (2024e) demonstrates near-linear scaling in long-context prefill latency through two approaches: Pass-KV, which transfers Key and Value matrices between GPUs for KV cache reuse, and Pass-Q, which transfers only Query matrices to reduce bandwidth and latency during decoding. For further exploration of how recent research has enhanced the efficiency of distributed attention, please refer to Section 5 and 6.

7.3.2 Scheduling Strategies of Inference Service

Currently, inference service providers face two key challenges (Sun et al., 2024b): the unpredictable nature of input lengths and the lack of effective scheduling strategies. As the demand for processing long texts continues to grow, the variability in input lengths has expanded, further complicating the situation (Patel et al., 2024b). Without proper scheduling

strategies, inference systems using traditional tensor, pipeline, and data parallelism alone would be less efficient at large cluster scales (Guo et al., 2024a).

Disaggregated Inference The prefill and decoding stage of LLM inference have fundamentally different characteristics and resource requirements (Raman, 2024; Patel et al., 2024c; Qin et al., 2024a):

- prefill is computationally intensive with its superlinear scaling with batch size and sequence length. Time to First Token (TTFT) is an important metric for this stage.
- decoding is memory(bandwidth)-constrained with its sublinear scaling with batch size. Time Between Tokens (TBT) and end-to-end latency are key metrics.

Given these differences, disaggregating the two stages (Patel et al., 2024c; Zhong et al., 2024c; Qin et al., 2024a; Hu et al., 2024c; Jin et al., 2024b) enables targeted optimization of tasks with two distinct computational characteristics, balancing computational efficiency, memory utilization, and latency requirements through independent resource pools and scheduling strategies, improving both latency and throughput.

Other Resource Partitioning & Scheduling Methods Several innovative approaches have been proposed for resource partitioning and scheduling in LLM inference (Lin et al., 2024d; Hu et al., 2024b; Lin et al., 2024b; Srivatsa et al., 2024; Wu et al., 2024a) . Infinite-LLM (Lin et al., 2024b) allows the independent scheduling and resource allocation for non-attention layers and improves system scalability through a two-tier global and local scheduling strategy. Co-optimizing KV state reuse and computation load-balancing, Preble (Srivatsa et al., 2024) is the first distributed LLM serving platform that targets prompt sharing. Elastic Sequence Parallelism (Wu et al., 2024a) dynamically adjusts to resource usage variations for prefill and decode stages, reducing KV cache migration overhead and fragmentation.

Multi-level cache management has emerged as another key optimization strategy, with several studies (Jiang et al., 2024b; Qin et al., 2024a; Song et al., 2024c; DeepSeek-AI, 2024) utilizing hierarchical distributed caches across GPUs, CPUs, DRAM, and SSDs. These studies implement load-aware scheduling and pre-estimate input/output lengths to optimize resource utilization.

Open Source Frameworks

Open-source frameworks have proven effective for handling context lengths of up to 100k tokens. More recent frameworks have optimized sequence processing through structured output (Zheng et al., 2024c) and cache reuse while maintaining high throughput (Kwon et al., 2023; Zheng et al., 2024c). Organizations and famous enterprises have also released open-source inference frameworks (Qin et al., 2024a; NVIDIA, 2024b; Zhihu & ModelBest Inc., 2024; Contributors, 2023; Hugging Face, 2024), each offering unique features. The accumulated engineering expertise from these projects has enriched technical options and advanced the field toward maturity.

vLLM Developed by the University of California, Berkeley, vLLM (Kwon et al., 2023) is renowned for its PagedAttention mechanism and strong open-source community support. It supports a wide range of models, including multimodal and non-Transformer architectures, and is compatible with diverse hardware. The upcoming version 1.0 will address previous limitations such as reliance on serial scheduling, limited graph optimization, and complex codebases that hinder further development.

SGLang Also from UC Berkeley, SGLang (Zheng et al., 2024c) is primarily written in Python and optimized with the torch.compile tool. It features optimizations like Radix Attention, structured output enhancements, and multi-process GMP transmission, which significantly reduce CPU overhead.

LMDeploy Developed by SenseTime and the Shanghai AI Laboratory, LMDeploy (Contributors, 2023) provides implementations based on both CUDA and Triton acceleration.

This framework supports multimodal tasks effectively and includes several commonly used pre-trained models.

Huggingface's Text Generation Inference Huggingface's Text Generation Inference (TGI) (Hugging Face, 2024) also utilizes the PagedAttention mechanism and employs Rust for low-level functions and Python (70%) for higher-level layers. Despite this, its throughput performance is average, particularly with larger batch sizes, due to decreased GPU memory management efficiency. Additionally, its CPU-GPU serial scheduling design limits GPU resource utilization.

TensorRT-LLM TensorRT-LLM (NVIDIA, 2024b) is NVIDIA's open-source framework on their GPUs. The framework stands out for its comprehensive optimization of popular LLMs, multiple NVIDIA hardware platforms(H100, L40, A100, V100, T4, etc.), flexible customization of plugins and kernels, and seamless multi-GPU/multi-node deployment capabilities.

8 Long-Context Pre-training

The development of deployment and training infrastructure has enabled the training and inference of LLMs with longer contexts. In this background, the pre-training length of LLMs has evolved from the initial 2k tokens (Touvron et al., 2023a) to 4k (Touvron et al., 2023b), 32k (Xiao et al., 2024c; Cai et al., 2024c), over 128k (Meta, 2024a; InternLM, 2024), and even 1M (Liu et al., 2024e). To expand the context length of LLMs effectively, more training strategies specialized for long-context LLMs are necessary. We begin our analysis from the long-context pre-training. Compared to the preceding short-context pre-training, long-context pre-training is featured with requiring fewer tokens, generally 1B-10B, and facing both challenges of quality and quantity (Fu et al., 2024b; Lv et al., 2024a).

8.1 Long-Context Data Quality

In the earliest works, researchers often focused on the length of pre-training (Chen et al., 2023b; Roziere et al., 2023; Peng et al., 2024b), with little discussion of other factors. Subsequently, ScalingRoPE first discovers that continual pre-training at the original pre-training context length could extrapolate the context length of LLMs (Liu et al., 2024p). LLaMA2Long (Xiong et al., 2024a) further points out that in long-context pre-training, data quality is more crucial than data length and provides detailed discussions on the mixing ratio and training cycles between long and short data.

Following this, Fu et al. (2024b) first raises the concept of long-context data engineering and suggests that the data required for long-context training is much less than that for short-context pre-training. Only 0.5B to 5B tokens are enough. Instead of relying solely on long book and long paper data, Fu et al. (2024b) also emphasizes that, besides length up-sampling, it is essential to maintain balance across domains, which has gained widespread acceptance (Zhang et al., 2024l; Young et al., 2024; ChatGLM, 2024; Gao et al., 2024d). Recently, Gao et al. (2024d) conducts an in-depth investigation into long-context training, finding that mixing code repositories and long books with high-quality short-context data is crucial for both long-context performance and retaining the short-context capabilities. The exploration of long-short-mixing training inspires thinking about training long-context LLMs from scratch, which will be discussed in **Q7** in Section12

Regarding the quality of a single long data sample, LongWanjuan (Lv et al., 2024a) is the first to propose that using LLM-based or rule-based metrics could reflect whether a long text exhibits long-context dependency characteristics from the perspective of coherence, cohesion, and diversity. It then categorizes long texts into holistic, aggregated, and chaotic types and conducts data mixing to achieve optimal long-context training results. ProLong (Chen et al., 2024c) goes deeper into long-context dependencies, designing scores for dependency strength, dependency distance, and dependency specificity to measure long-distance dependencies between different segments in a long text, for data filtering.

8.2 Long-Context Data Curation

Discussions on long-context data quality remain very limited, primarily because long-context data itself is extremely scarce, leading to a greater focus on data synthesis (ChatGLM, 2024). In early long-context training, researchers employ the simplest splicing methods to obtain sufficient long-context data (Chen et al., 2024i; Tworkowski et al., 2024; Chen et al., 2024a; Li et al., 2024h). Notably, CodeLLaMA utilized the feature of code data to concatenate code from the same project, resulting in ultra-long code datasets (Roziere et al., 2023).

Subsequent efforts begin to stitch similar short texts into a long context through similarity matching. For instance, ICLM (Shi et al., 2024) constructs a graph of documents with embeddings from an encoder-only model and applies the traveling salesman algorithm to extract efficiently. SPLiCe (Staniszewski et al., 2023) replaces selection criteria with BM25 retrieval or attribute label matching and extends the splicing length to 32k. BM25Chunk (Zhao et al., 2024e) provides in-depth analysis for training on concatenated long-context data, while later work explored retrieval methods using LLM embeddings (ChatGLM, 2024) and keyword matching (Gao et al., 2024b). DataSculpt attempted to optimize the synthesis of spliced data through multi-objective combinatorial optimization (Lu et al., 2024a).

In addition to sequential splicing, a few works have attempted to achieve extended length through interleaved splicing of short texts (Zhao et al., 2024b; Tian et al., 2024). LongSkywork proposes CIP (Zhao et al., 2024b), which splits, shuffles, and splices short texts, allowing LLMs to identify relevant segments within seemingly chaotic contexts through self-attention adaptively, thus enhancing long-context modeling capabilities. Following this, UTK (Tian et al., 2024) introduces knot tokens pushing LLMs to untie these knots and gain long-context capabilities more effectively. These methods could significantly improve the performance of synthetic tasks such as RULER (Hsieh et al., 2024a).

Additionally, a few pieces of research concern loss design specialized for long-context training (Fang et al., 2024b). Discussions regarding long-context pre-training work are still limited, which we will highlight and summarize in **Q8** in Section12, and much of the discourse is dispersed across various technical reports of LLMs. We have compiled these technical reports of long-context LLMs, listing the information related to long-context pre-training, post-training, and evaluation, for the reader's reference.

Model	Organization	Time	Version	Context Length	Benchmark
ChatGPT (2022)	OpenAI	22.11	gpt-3.5-turbo gpt-3.5-turbo-instruct gpt-3.5-turbo-0125	4K 4K 16K	-
GPT-4 (2023a)	OpenAI	23.03	(default) turbo	128K	-
GPT-4o (2023a)	OpenAI	24.05	(default) mini	128K	-
OpenAI-o1 (2024)	OpenAI	24.09	(default) mini	200K 128K	-
Claude (2023)	Anthropic	23.03	(default)	-	-
Claude2 (2024a)	Anthropic	23.07	(default) 2.1	100K 200K	-
Claude3 (2024b)	Anthropic	24.03	Haiku Sonnet Opus	200K	NIAH
Claude3.5 (2024b)	Anthropic	24.06	Haiku Sonnet Opus	200K	-
Gemini (2023)	Google	23.12	Ultra Pro Nano	32K	SCROLLS
Gemini-1.5 (2024)	Google	24.02	Pro Flash	1M	NIAH, LQA, LICL
Gemini-2.0 (2024)	Google	24.12	Pro Flash	1M	LQA
DeepSeek-R1 (2024)	DeepSeek	24.11	Lite-Preview	-	-
Kimi-chat (2023)	MoonshotAI	23.11	(default)	2M	NIAH
AFM (2024)	Apple	24.07	(default)	32k	LQA
abab (2024)	MiniMax	24.04	6.5s 7	240k	NIAH
Step-1 (2024b)	Step	24.03	(default)	256k	-
Step-2 (2024b)	Step	24.07	(default)	16k	-

Table 1: Comparison of mainstream close-source long-context LLMs. The symbol "-" indicates that no relevant information was found. *Benchmark* refers to the long-context benchmarks used in the evaluation. Specifically, *PPL* stands for perplexity, *LQA* for Long QA, *LC* for Long Code, and *LICL* for Long In-Context Learning.

Model	Organization	Time	Version	Architecture Detail (Base-Q-KV)	Context Length	Pre-Training Strategy	Post-Training Strategy	Benchmark
LLaMA (2023a)	Meta	23.03	7B 13B 33B 65B	1e4-32Q-32KV 1e4-40Q-40KV 1e4-52Q-52KV 1e4-64Q-64KV	2k	len=2k	-	-
LLaMA2 (2023b)	Meta	23.07	7B 13B 70B	1e4-32Q-32KV 1e4-40Q-40KV 1e4-64Q-8KV	4k	len=4k	-	SCROLLS
LLaMA3 (2024a)	Meta	24.04	8B 70B	5e5-32Q-32KV 5e5-64Q-8KV	8k	len=8k	-	-
LLaMA3.1$^{\diamond}$ (2024)	Meta	24.07	8B 70B 405B	5e5-32Q-8KV freq 1,4; factor 8 5e5-64Q-8KV freq 1,4; factor 8 5e5-128Q-8KV freq 1,4; factor 8	128k	len=8k→128k; context parallelism	Iterative training; synthetic data	LQA, LICL, ZeroSCROLLS, NIAH, InfiniteBench
LLaMA3.2$^{\diamond}$ (2024b)	Meta	24.09	1B 3B 11B	5e5-32Q-8KV freq 1,4; factor 32 5e5-32Q-8KV freq 1,4; factor 32 5e5-32Q-8KV freq 1,4; factor 8	128k	-	-	-
LLaMA3.3$^{\diamond}$ (2024)	Meta	24.12	70B	5e5-64Q-8KV freq 1,4; factor 8	128k	-	-	-
Gemma (2024a)	Google	24.03	2B 7B	1e4-8Q-1KV 1e4-16Q-16KV	8k	len=8k	-	-
Gemma2$^{\flat}$ (2024b)	Google	24.06	3B 9B 27B	1e4-8Q-4KV Sliding Window=4096 1e4-16Q-8KV Sliding Window=4096 1e4-32Q-16KV Sliding Window=4096	8k	len=8k	-	-
Mistral-v0.1$^{\flat}$ (2023a)	MistralAI	23.1	7B	1e4-32Q-8KV Sliding Window=4096	8k	-	-	-
Mistral-v0.2 (2023a)	MistralAI	23.11	7B	1e6-32Q-8KV	32k	-	-	-
Mistral-v0.3 (2023a)	MistralAI	24.1	7B	1e6-32Q-8KV	32k	-	-	-

Table 2: Comparison of mainstream open-source long-context LLMs. The symbol "-" indicates that no relevant information was found. *Architecture Details* is composed of *Base-Q-KV*, which respectively represent the RoPE Base, num_attention_heads and num_kv_heads. If RoPE is not used, the type of positional encoding employed will be specified in the *RoPE Base* field. The symbol "$\diamond$" indicates that Scaling RoPE is used and we provide the scaling frequency and scaling factor below the *Base-Q-KV*. The symbol "$\flat$" indicates that Sliding Window Attention is used and we provide the sliding window below the *Base-Q-KV*. *Context Length* refers to the maximum length of context that the model can process. *Pre-Training Strategy* and *Post-Training Strategy* refer to the strategies employed by the model for handling long contexts during the respective pre-training and post-training phases. Additionally, we provide the context lengths (denoted as *len*) used during long-context training, as specified in the technical reports. *Benchmark* refers to the long-context benchmarks used in the evaluation. Specifically, *PPL* stands for perplexity, *LQA* for Long QA, *LC* for Long Code, and *LICL* for Long In-Context Learning.

Model	Organization	Time	Version	Architecture Detail (Base-Q-KV)	Context Length	Pre-Training Strategy	Post-Training Strategy	Benchmark
phi-3 (2024a)	Microsoft	24.04	Phi-3.5-MoE Phi-3.5-Mini	1e4-32Q-8KV 1e4-32Q-32KV	128k	Long-RoPE	-	RULER, LC
phi-4 (2024b)	Microsoft	24.12	Phi-4-14B	2.5e5-40Q-10KV	16k	len=4k; Mix long and short context	-	HELMET
Falcon (2023)	TII	23.11	7B 40B 180B	1e4-71Q-1KV 1e4-128Q-8KV 1e4-232Q-8KV	2k	len=2k	-	-
Falcon2 (2024)	TII	24.07	11B	5e5*-32Q-8KV	8k	len=2k→8k	-	-
Falcon3 (2024c)	TII	23.12	1B 3B 7B 10B	1e6*-8Q-4KV 1e6*-12Q-4KV 1e6*-12Q-4KV 1e6*-12Q-4KV	4k 8k 32k 32k	-	-	-
Qwen (2023a)	Alibaba	23.09	1.8B 7B 14B 72B	1e4-16Q-16KV 1e4-32Q-32KV 1e4-40Q-40KV 1e6-64Q-64KV	8k 32k 32k 32k	len=2k; NTK	-	PPL
Qwen1.5 (2023a)	Alibaba	24.02	0.5B 1.8B 4B 7B 14B 32B 72B	1e6-16Q-16KV 1e6-16Q-16KV 5e6-20Q-20KV 1e6-32Q-32KV 1e6-40Q-40KV 1e6-40Q-8KV 1e6-64Q-64KV	32k	len=32k	-	L-Eval
Qwen2 (2024a)	Alibaba	24.07	0.5B 1.5B 7B 72B	1e6-14Q-2KV 1e6-12Q-2KV 1e6-28Q-4KV 1e6-64Q-8KV	128k	len=4k→32k; RoPE base=1e4→1e6; YaRN, DCA	-	NIAH, NeedleBench, LV-Eval
Qwen2.5 (2024)	Alibaba	24.09	0.5B 1.5B 3B 7B 14B 32B 72B	1e6-14Q-2KV 1e6-12Q-2KV 1e6-16Q-2KV 1e6-28Q-4KV 1e6-40Q-8KV 1e6-40Q-8KV 1e6-64Q-8KV	128k 128k 128k 128k 128k 128k 128k	len=32k→256k; RoPE base=1e4→1e6; YaRN, DCA	len=32k→256k	RULER, LV-Eval, LongBench-chat
QwQ (2024a)	Alibaba	24.11	32B-preview	1e6-40Q-8KV	32k	-	-	-
Index (2024)	Bilibili	24.10	1.9B	3.2e6-16Q-16KV	32k	len=32k; Doc Packing	len=32k; Long SFT; Doc Packing	NIAH, LongBench, LEval
MiniMax-01♮ (2025)	MiniMax	25.01	Text-01	1e7-64Q-8KV	4M	len=32k→1M	len=8k→1M; Long SFT and Long RL	NIAH, RULER, LongBench-v2, MTOB

Table 3: Continued table of Table 2. The symbol * indicates that the actual RoPE Base is the annotated value plus 42. The symbol ♮ indicates that lightning attention is used.

Model	Organization	Time	Version	Architecture Detail (Base-Q-KV)	Context Length	Pre-Training Strategy	Post-Training Strategy	Benchmark
DeepSeek-V2 (2024a)	DeepSeek-AI	24.05	(default) Lite	1e4-128Q-128KV† 1e4-16Q-16KV†	128k	len=32k; YaRN	-	NIAH
DeepSeek-V2.5 (2024a)	DeepSeek-AI	24.08	(default)	1e4-128Q-128KV†	128k	len=32k; YaRN	-	NIAH
DeepSeek-V3 (2024)	DeepSeek-AI	24.12	(default)	1e4-128Q-128KV†	128k	len=32k→128k; YaRN	Distill long-CoT capacity from DeepSeek-R1	LongBench-v2, LQA
ChatGLM (2024)	Zhipu, THU	23.05	6B	1e4-32Q-32KV	2k	len=2k	-	-
ChatGLM2 (2024)	Zhipu, THU	23.06	6B	1e4-32Q-16KV	32k	-	len=32k; long SFT; Positional Interpolation	
ChatGLM3 (2024)	Zhipu, THU	23.1	6B	1e4-32Q-16KV	32k	-	-	-
GLM-4 (2024)	Zhipu, THU	24.06	9B 9B-chat 9B-chat-1M	1e4-32Q-16KV	8k 128k 1M	len=8k→1M	LongAlign; multi-task long SFT	LongBench-chat
InternLM2 (2024c)	InternLM	23.12	1.8B 7B 20B	1e6-16Q-8KV 1e6-32Q-8KV 1e6-48Q-8KV	200k	len=4k→32k; NTK; RoPE base=5e4→1e6;	len=32k; long SFT; book and code data	L-Eval, LongBench, NIAH
InternLM2.5 (2024)	InternLM	24.08	1.8B 7B 20B	1e6-16Q-8KV 5e7-32Q-8KV 5e7-48Q-8KV	1M	len=1M	-	-
InternLM3◇ (2025)	InternLM	25.01	8B	5e7-32Q-2KV factor 6	1M	-	-	RULER
Yi (2024)	01.AI	23.11	6B 9B 34B	5e6-32Q-4KV 1e7-32Q-4KV 1e7-56Q-8KV	200k 200k 200k	len=4k; NTK from 4k to 200k; book and synthetic data	Long SFT; synthetic data	NIAH
Yi-1.5 (2024)	01.AI	24.05	6B 9B 34B	5e6-32Q-4KV 5e6-32Q-4KV 5e6-56Q-8KV	4k 32k 32k	-	-	-
Baichuan (2023)	Baichuan-Inc	23.06	7B 13B	1e4-32Q-32KV 1e4-40Q-40KV	4k	-	-	-
Baichuan2 (2023)	Baichuan-Inc	23.09	7B 13B	1e4-32Q-32KV ALiBi-40Q-40KV	4k	-	-	-
MiniCPM (2024g)	OpenBMB	24.02	2B	1e5-36Q-36KV	4k	-	-	-
MiniCPM2 (2024g)	OpenBMB	24.04	1B 2B	1e5-24Q-24KV 1e6-36Q-36KV	4k 128k	len=4k→128k	Long SFT; synthetic long QA data	InfiniteBench
MiniCPM3 (2024g)	OpenBMB	24.08	4B	1e5-40Q-40KV	32k	Long-RoPE	-	-

Table 4: Continued table of Table 2. The symbol † indicates that MLA is used in this model.

9 Long-Context Post-training

Based on the above long-context pre-training strategy, long-context LLMs are trained to understand the long context well. Subsequently, the post-training is introduced to ensure the LLMs can follow human instructions and preferences are problems that need to be addressed during post-training (Dubey et al., 2024; Bai et al., 2024a). Long-context post-training methods can be classified into three categories based on the length of input and output: ***Long-In-Short-Out***, ***Short-In-Long-Out***, and ***Long-In-Long-Out***. Currently, there is a lack of research on Long-In-Long-Out, which is an important direction for future studies. Therefore, we will focus the following discussion on the Long-In-Short-Out and Short-In-Long-Out scenarios and add something beyond post-training later.

9.1 Long-In-Short-Out

In the post-training process of LLMs, task-specific data is typically constructed to enhance the LLM's performance on particular tasks, with the data construction type determined by the method (Supervised Fine-Tuning, SFT or Reinforce Learning RL). In the Long-In-Short-Out scenario, due to the length of the input context, manual annotation is difficult, and thus, synthetic data is often used. This section will introduce common data construction methods for various tasks.

- **Instruction Following** Provided with long-context data, instructions are given to generate relevant responses (Chen et al., 2024i), or prompts are used to guide the LLMs to generate corresponding instructions and responses (Köksal et al., 2023; Bai et al., 2024a).
- **DocQA** Given a long document, relevant questions and answers are generated using LLMs. These can be based on the entire document (Kaili-May Liu et al., 2024), or on specific context segment (An et al., 2024c; Xiong et al., 2024a; Dubey et al., 2024). In some cases, questions are generated and information is retrieved to ensure the quality of the generated answers (Anonymous, 2024; Yu et al., 2023). Some researchers use shorter context segments to construct QA pairs and then concatenate many short pieces to form a long document (Li et al., NA; Young et al., 2024). To ensure the quality of responses, LLMs are often asked to provide citations (Zhang et al., 2024d).
- **Multi-Hop QA** Long-context multi-hop QA data can usually be formed by combining multiple single-hop QA data (Trivedi et al., 2022). When combining single-hop QAs, similarity or question relevance can be considered (Chen et al., 2024k) to ensure coherence in question generation. Some studies require LLMs to generate responses using methods such as CoT (Wei et al., 2022) or citation to improve data quality (Li et al., 2024m).
- **Summarization** Besides using manually written documents and summary data from the Internet, LLMs are also often used to summarize long contexts. One method is to split the long context into chunks and summarize them individually, then provide a final summary of the summaries (Dubey et al., 2024; ChatGLM, 2024). Another method is to summarize short contents first, then concatenate them into a longer document and summarize that (Li et al., NA).
- **Retrieve** Information is inserted into the long context, and questions are posed about the inserted information (Kamradt, 2023). Alternatively, several pieces of information are combined to create a long context, and a question is asked about specific information (Xiong et al., 2024b).

Researchers have also studied how to filter data. LOGO (Tang et al., 2024c) scores the contribution of answers from different chunks to determine the quality of data samples. LongReward (Zhang et al., 2024e) uses predefined rules to score the responses. GATEAU (Si et al., 2024) filters data based on the relevance of the final reply to the long document and the importance of certain parts in the response, giving high attention weights to crucial parts. This method has shown significant effects with only a small amount of data.

Some researchers have explored methods to improve long-context capabilities without constructing long-context data. SkipAlign (Wu et al., 2024h) modifies the position embedding indices in short-text data, training LLMs on short texts to give it the ability to handle long texts. ProLong (Gao et al., 2024d) adjusts the data sources and proportions of long and short texts to find efficient long-context LLM training methods. It has been found that using only short-text instruction data can also help the LLMs perform well on long-text tasks.

9.2 Short-In-Long-Out

When the task is more complex or requires detailed steps, longer output is necessary to express thoughts and details (Wei et al., 2022; Yao et al., 2024c). Therefore, long output is also a key capability for long-context LLMs. Data construction in this field is challenging, and there is still insufficient research. Current data construction methods can be classified into three categories: backtranslation, planning, and iterative training.

Backtranslation In backtranslation, given the context and a response, the LLMs generates instruction data in reverse. This method leverages the long-context LLMs' strong ability to handle long inputs (Pham et al., 2024).

Planning Another method is planning, which involves breaking the task down into subtasks to reduce complexity, eventually solving the original task. Some researchers apply planning by breaking down writing tasks, first generating an outline and then using the outline to create segments that combine into the final text (Bai et al., 2024c; Liang et al., 2024d). Li et al. (2024i) also uses planning to guide reasoning, improving LLMs' reasoning.

Iterative Training Iterative training is also a commonly used method for enhancing LLMs' capabilities in the post-training stage. Self-Lengthen (Quan et al., 2024) uses two LLMs, a Generator and an Extender. The Generator generates responses within a specified length range, and the Extender extends the content to the target length. The concatenated extended data is then used to train the next generation of Generator and Extender.

Long Thought Long-output tasks, especially long thought, have attracted particular attention. The success of generation strategies like CoT (Wei et al., 2022) and ToT (Yao et al., 2024c) has shown that LLMs can fully utilize their reasoning capabilities to generate better results. OpenAI o1 (OpenAI, 2024) further enhances reasoning ability with CoT (Wei et al., 2022), achieving impressive results. More and more research is focused on how to achieve o1 or even better performance in long-context reasoning (Zeng et al., 2024a). Qin et al. (2024b); Huang et al. (2024c) improves the LLMs' reasoning ability using tree search and multi-agent strategies. ConTReGen (Roy et al., 2024) applies planning strategies in document QA by first generating sub-tasks from top-down and then retrieving relevant documents to solve the sub-tasks until the entire task is completed. Long thought is an important task and we will discuss it in **Q9** in Section12

9.3 Beyond Post-Training

Besides post-training, many methods are being explored to enhance long-context LLMs. ***Test Time Training*** (TTT) utilizes self-supervised learning during inference to further train LLMs using input test data (Sun et al., 2020; Liang et al., 2024b). Temp-Lora (Wang et al., 2024m) applies TTT in long-context scenarios by fine-tuning temporary Lora modules using contextual information during inference, guiding generation. Some works achieve alignment by providing examples or guidance during inference (Sun et al., 2024h; Zhang et al., 2024i; Xie et al., 2023), and long-context LLMs facilitate the effectiveness of these methods. Some researchers have effectively improved the performance of LLMs through ***Test-Time Scaling*** (Liao & Vargas, 2024; Snell et al., 2024), proposing a new direction and further emphasizing the importance of long context. Additionally, LUQ (Zhang et al., 2024a) focuses on calibration for long-context LLMs, using NLI classifiers to determine the confidence of generated results and reducing uncertainty through model ensembling.

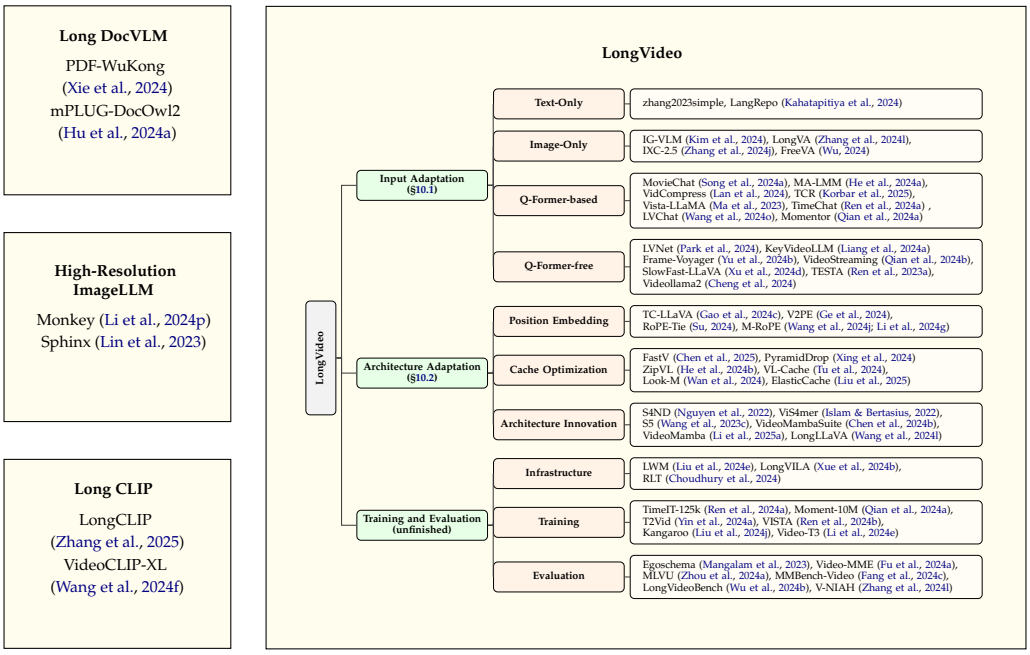

Figure 10: An overview of the long context in Multi-modal LLMs.

10 Long-Context MLLM

Based on the above technique, now we have numerous LLMs with strong long-context capabilities. But that is not the end. Long context is crucial for LLM focused on textual information and holds even greater significance for Multi-modal LLM (MLLM). In this section, we will change our focus on Long-context MLLMs. Long-context MLLMs involve various scenarios, including DocVLM for long documents with images (Xie et al., 2024; Hu et al., 2024a), ImageLLM for high-resolution images (Li et al., 2024p; Lin et al., 2023), VideoLLM for long videos (Zhang et al., 2024l; Wang et al., 2024j), CLIP with long descriptions (Zhang et al., 2025; Wu et al., 2024g), as well as long speech models (Reid et al., 2024) and even world models(Liu et al., 2024e; Zhan et al., 2024). In this version, we only discuss long VideoLLM in detail. We omit speech for it is 1D like text and can be viewed as a new language (Zhang et al., 2023b; 2024b). Regarding long DocVLM, while it matters practically (Jaisankar et al., 2024; Zhang et al., 2024p; Ma et al., 2024b; Chia et al., 2024), related discussions are relatively limited (Xie et al., 2024; Hu et al., 2024a; Blau et al., 2024; Liu et al., 2024c).

Long CLIP and ImageLLM emphasize the extension of descriptive texts (Zheng et al., 2025; Zhang et al., 2025; Wang et al., 2024f) and images (Li et al., 2024p; Lin et al., 2023), respectively. On one hand, the study of long CLIP could generally follow a length extrapolation discussion in text domain (Zhang et al., 2025; Najdenkoska et al., 2024). On the other hand, while currently viewed as a long-context issue, high-resolution ImageLLM faces the backbone in the design of the vision encoder and may not necessarily remain a long-context problem in the future. In contrast, Long VideoLLM presents the most discussed long-context challenges in the MLLM due to its rich fine-grained spatiotemporal details as well as long-term dependencies in long videos (Zou et al., 2024a; Li et al., 2024g). Therefore, the following content will focus on the issues related to Long Video in detail.

Generally, long VideoLLMs and other long-context MLLMs training are after the textual pre-training and fine-tuning (Liu et al., 2024e; Zhang et al., 2024l). Accordingly, this section focuses on how to obtain a long-context MLLM, especially a long VideoLLM, from a trained long-context LLM through input and architecture adaptation as well as multi-modal training. We will also discuss long video evaluations at the end of this section. This section can be viewed as a microcosm of the entire survey, emphasizing the differences and extensions of long-context in videos compared to text. Unlike text, videos have lower information density

and are characterized by sample extraction (Zou et al., 2024a). Many studies concentrate on the extraction and compression of video information (Song et al., 2024a; Ren et al., 2024a; Qian et al., 2024a; Yu et al., 2024b). However, discussions on fine-grained alignment from long texts to long videos, such as the generalization of position embedding(Su, 2024; Wang et al., 2024j) and reasoning capabilities(Li et al., 2024e), are relatively scarce.

10.1 Input Adaptation

The adaptation of long-context LLMs to long videos begins with input processing. Unlike text, which can be directly tokenized, video requires frame sampling, patch segmentation, and vision encoding and connector processing before entering the LLM (Zou et al., 2024a). For example, a one-hour video, sampled at 2 frames per second (fps) with 66 tokens per frame (Wang et al., 2024j), results in over 400k tokens, while sampling at 1 fps with 2 tokens per frame (Li et al., 2025b) yields fewer than 8k tokens. Thus, the adaptation of video input affects the MLLM's context length, processing efficiency, and downstream performance. The input adaptation of long VideoLLMs has benefited from the redundancy of video (Zou et al., 2024a), utilizing both text-only (Zhang et al., 2023a) and image-only (Kim et al., 2024; Zhang et al., 2024l) methods to replace original video inputs, as well as employing modules such as Q-Former (Li et al., 2023c; Ma et al., 2023) to compress video data.

Text-Only A relatively simple method for long video input is to truncate the long video into several short segments, convert them into corresponding text descriptions, concatenate these to form a complete video description and input it to the LLMs (Zhang et al., 2023a). This approach was first proposed in LLoVi (Zhang et al., 2023a) and later improved by LangRepo (Kahatapitiya et al., 2024), which iteratively processes video segments along with previous descriptions, eventually to generate a description for the entire video. Similarly, MVU (Ranasinghe et al., 2024) further simplifies the video into a combination of key information about objects. These methods do not input the original long video into the backbone LLM, thus reducing the adaptation and processing costs. Subsequent work has combined this approach with agents, enabling LLMs to collaborate with VLMs (Wang et al., 2025) or interactively invoke tools (Fan et al., 2025) for long video understanding.

Image-Only Another class of methods treats videos as a comic strip, allowing LLMs to adapt to image features without additional long-video training. This line of work can be traced back to IG-VLM (Kim et al., 2024), which achieves video understanding by transforming a video into a single image by arranging multiple frames in a grid. InternLM2-XComposer2.5 (Zhang et al., 2024j) (IXC-2.5 for simplicity) inherits this method and exhibits strong performance on various video benchmarks. Meanwhile, FreeVA (Wu, 2024) also demonstrates that using a similar approach without video training conditions enables ImageLLMs to process video. LongVA (Zhang et al., 2024l) further proposes that the image-only method can transform an ImageLLM with long-context capabilities into a long VideoLLM capable of handling 2000 frames or over 200k visual tokens.

Q-Former-based Besides the tricks mentioned above, early VideoLLMs tend to use cross-attention-based Q-Former (Li et al., 2023c; Zhang et al., 2023c; Song et al., 2024a) to compress multi-modal information into fixed-length inputs, due to the significant redundancy in video representations (Zou et al., 2024a). However, in the scenario of long videos, Q-Former faces greater challenges in processing capacity, which introduce techniques including memory (Song et al., 2024a), keyframe selection (Korbar et al., 2025), Q-Former variants (Ma et al., 2023; Ren et al., 2024a), and timestamp enbedding (Ren et al., 2024a; Qian et al., 2024a).

MovieChat (Song et al., 2024a) first introduces the memory mechanism into long VideoLLM, using a queue-based short-term memory and long-term memory based on adjacent fusion with higher frame similarity. Similarly, the concepts of compression and memory are also referenced in MA-LMM (He et al., 2024a) and VidCompress (Lan et al., 2024). Besides compression and memory, keyframe extraction is also an intuitive approach. For instance, TCR (Korbar et al., 2025) locates relevant information based on text and feeds it to Q-Former. TGB (Wang et al., 2024p) uses the RoPE-involved product between optical flow and text embeddings to identify the ranges of multiple key content segments. Furthermore, some

works attempt to overcome the fixed-length output constraint of Q-Former. For example, Vista-LLaMA (Ma et al., 2023) proposes a SeqQ-Former similar to Transformer-XL (Dai et al., 2019), and TimeChat (Ren et al., 2024a) introduces a sliding window-based Q-Former. Notably, in LVChat (Wang et al., 2024o), long videos are interleaved into multiple groups, encoded separately, and interleaved reversely to achieve complete encoding within a limit.

Although Q-Former faces limitations in processing long videos, the attention in Q-Former still facilitates the injection of information beyond images, particularly temporal and spatial embeddings (Ren et al., 2024a; Qian et al., 2024a). For example, TimeChat (Ren et al., 2024a) employs a timestamp-aware frame encoder that explicitly binds visual content with corresponding timestamps in Q-Former with prompts. Similarly, TCR (Korbar et al., 2025) injects temporal information into visual representations through prompts based on special tokens after keyframe extraction. Momentor (Qian et al., 2024a) achieves the same goal with temporal embeddings in a continuous temporal token space. Some Q-Former-free models also adopt recurrent compression (Wang et al., 2024q) and temporal encoding. For instance, VideoStreaming (Qian et al., 2024b) uses structures similar to RMT (Bulatov et al., 2022) and LandmarkAttention (Mohtashami & Jaggi, 2023) to recurrently encode images, injecting corresponding temporal information with text prompts during the encoding process and recalling only relevant vision encoding segments during inference.

Q-Former-Free In MLLMs, the LLaVA series (Liu et al., 2024h;f;g; Zhang et al., 2024u) first propose feeding the visual tokens from the vision encoder directly to the LLM backbone, avoiding the information bottleneck caused by Q-Former, which is inherited by many subsequent researches (Li et al., 2024b; Wang et al., 2024j; Zhang et al., 2024j). However, due to the high redundancy of video information, long VideoLLMs also introduce keyframe extraction and token compression to balance the compression of redundant information with the retention of key information (Zou et al., 2024a; Yu et al., 2024b; Cheng et al., 2024).

Regarding keyframe extraction, early attempts are often limited in uniform sampling of long videos (Zhang et al., 2024u; Cheng et al., 2024), resulting in low efficiency and high information loss (Shen et al., 2024), prompting subsequent improvements. For instance, LVNet (Park et al., 2024) enhances keyframe extraction efficiency through a hierarchical keyframe selector, while KeyVideoLLM (Liang et al., 2024a) employs frame clustering to find central frames and integrates keyframe extraction in instruction fine-tuning. Frame-Voyager (Yu et al., 2024b) trains a frame extraction module by enumerating all possible frame extractions, discovering that extracting only 8 keyframes can achieve good understanding.

Regarding token compression, unlike text-only LLMs, which are limited to uni-modal, uni-dimensional, and unreadable compression, VideoLLM compression explores the differences in information density between image and text information, as well as how to integrate spatiotemporal information better. For the former, LLaMA-VID (Li et al., 2025b) leverages cross-attention between textual queries and visual features, arguing that one frame is worth two tokens in VideoLLM. SlowFast-LLaVA (Xu et al., 2024d) combines fine-grained slow features and coarse-grained fast features to achieve effective and efficient representation for detailed video understanding. For the latter, VideoLLM employs not only intuitive methods such as adjacent token merging in LLaVA-NeXT-Video (Zhang et al., 2024u) and LongLLaVA (Wang et al., 2024l), hierarchical token merging (Weng et al., 2025), and average pooling (Cai et al., 2024a), but also more exquisite techniques like adaptive pooling in PLLaVA (Xu et al., 2024c), temporal-spatial aggregation in TESTA (Ren et al., 2023a) and LongVU (Shen et al., 2024), and 3D convolution in Videollama2 (Cheng et al., 2024), Kangaroo (Liu et al., 2024j), Qwen2-VL (Wang et al., 2024j).

10.2 Model Adaptation

Position Embedding After inputting visual tokens into LLM, other problems arise, how to encode the relationship between visual tokens and textual tokens, and how to handle the extrapolation of visual tokens in the context of long videos (Su, 2024; Wang et al., 2024j; Li et al., 2024g). There are two schools of research regarding this. One ignores these questions or believes that VideoLLM can inherently perceive the spatiotemporal relationships of visual tokens without explicit representation in position embeddings (Liu et al., 2024e; Zhang

et al., 2024u; Chen et al., 2024j). For long videos, in addition to directly applying existing text extrapolation methods (Liu et al., 2024e; Zhang et al., 2024u; Shang et al., 2024) has also made some attempts. To avoid hallucinations caused by the increasing gap between video and text during generation, Vista-LLaMA (Ma et al., 2023) does not apply RoPE to image tokens. TC-LLaVA (Gao et al., 2024c) improves downstream performance by varying the growth steps of image and text token indices and applying full attention to the same frame images. V2PE (Ge et al., 2024) uses a similar approach to E^2-LLM (Liu et al., 2024i), employing variable and smaller increments for visual tokens to enable the model to handle 1M long sequences under a 256k training setting.

The other group of research argues that images and videos possess additional spatiotemporal features compared with text, necessitating a more sophisticated position embedding schema for explicit representation (Su, 2024; Wang et al., 2024j; Li et al., 2024g). For instance, Su (2024) first proposes RoPE-Tie and conducts a comprehensive mathematical analysis. Subsequently, Qwen2-VL (Wang et al., 2024j) introduced M-RoPE, which unifies the position embedding of text, image, and video by decomposing the feature dimensions of text from low to high dimensions to represent time, height, and width. Regarding extrapolation, Giraffe (Li et al., 2024g) proposes M-RoPE++ by combining the three split intervals with YaRN (Peng et al., 2024b) interpolation, achieving improved results.

Cache Optimization The redundancy of multi-modal information is reflected not only in the sampling or compressing of input content but also in the sparsity of attention distribution (Wan et al., 2024; Ma et al., 2023; Tu et al., 2024), which leads to the emergence of multi-modal cache optimization. However, since the compression of multi-modal information is more dominant in the input adaptation (Song et al., 2024a; Ren et al., 2024a; Qian et al., 2024a; Yu et al., 2024b), the exploration of KV cache optimization left for long VideoLLMs is relatively limited. Considering that KV cache optimization in the text domain has been discussed in Section 3, here, we primarily analyze the work on long video KV cache optimization by cache dropping and merging (Chen et al., 2025; Wan et al., 2024).

Regarding cache dropping, discussions in MLLMs are more centered on layer adaptation. For example, FastV (Chen et al., 2025) is the first to utilize LLMs' signal to guide the cache optimization, dropping visual tokens starting from the second layer of the MLLM in the inference phase. Similarly, PyramidDrop (Xing et al., 2024) emphasizes pruning more unimportant visual tokens as the layer goes up, and ZipVL (He et al., 2024b) presents a layer-wise adaptive dropping ratio that boosts the overall compression ratio and accuracy compared to a fixed ratio. Notably, VL-Cache (Tu et al., 2024) discovers that the attention patterns of MLLMs vary by modality, and therefore designing different sparsity levels for these patterns, adaptively adjusting the sparsity degree across layers.

Regarding cache merging, related explorations focus more on the input side (Li et al., 2025b; Shen et al., 2024; Lan et al., 2024), while there is less work on merging tokens within the attention block (Wan et al., 2024; Liu et al., 2025). Interestingly, Wan et al. (2024) finds that the attention score for textual tokens is very dense, whereas the attention score for visual tokens is sparse. However, this finding contradicts the results from Ma et al. (2023) and Tu et al. (2024), which show that the visual components are more attended.

Architecture Innovation In the text domain, the emergence of RWKV (Peng et al., 2023a; Choe et al., 2024) and SSM-Mamba (Gu et al., 2020; Gu & Dao, 2023; Dao & Gu, 2024) has led to new architectural innovations for LLMs, and similar research exists in the multi-modal field as well. Before the introduction of Mamba (Gu & Dao, 2023), S4ND (Nguyen et al., 2022), ViS4mer (Islam & Bertasius, 2022), and S5 (Wang et al., 2023c) utilized S4 (Gu et al., 2021a) blocks to capture long-context dependencies in video. After the introduction of Mamba (Gu & Dao, 2023), there have been efforts to model video using Mamba (Li et al., 2025a; Chen et al., 2024b). Besides, there are also long video hybrid architectures, including LongLLaVA (Wang et al., 2024l), and efficient attention approaches for long videos, such as VideoTree (Wang et al., 2024s).

11 Long-Context Evaluation

We finally come to the part of the long-context evaluation, which is an important technique of long-context LLM (An et al., 2023; Bai et al., 2023b; Zhang et al., 2024q; Kamradt, 2023; Hsieh et al., 2024a; Yen et al., 2024b). Before the mainstream length extrapolation methods emerged, long-context evaluation primarily includes four assessment methods. The first is language modeling perplexity, typically on datasets like WikiText (Merity et al., 2022) or PG19 (Rae et al., 2019). The second is Long-Range Arean (LRA) (Tay et al., 2021), testing whether models can capture the underlying structure through artificially constructed sequences. Furthermore, LongEval (Li et al., 2023a) assesses the retrieval ability of LLM across different context lengths through coarse-grained topic retrieval and fine-grained line retrieval. The only benchmark based on natural long texts to reflect the actual downstream performance is Scrolls (Shaham et al., 2022), along with its upgraded version ZeroScrolls (Shaham et al., 2023), which enrich the longer data samples in existing QA and summarization tasks.

With the development of long-context LLMs, researchers construct more benchmarks, as shown in Table 5 and 6. In this section, we will introduce the development of long-context evaluation from two perspectives, ***type of tasks*** and ***benchmark features***. In this process, we will reveal the pain point of long-context evaluation that persists from early explorations. If real texts are used to construct tasks, while they can reflect long-context scenarios more authentically, the evaluation length cannot scale automatically and a careful metric design is necessary (Zhang et al., 2023d; Xu et al., 2024f; Yen et al., 2024b). In contrast, if synthetic data are used, although lengths and metrics can be easily controlled, it is challenging to ensure that they are consistent with real-world scenarios (Hsieh et al., 2024a; Li et al., 2024f).

11.1 Type of Tasks

Long QA and Summary The evaluation of long-context LLMs originated from long QA and summarization. Early long-context benchmarks including Scrolls (Shaham et al., 2022), ZeroScrolls (Shaham et al., 2023), LEval (An et al., 2023), and LongBench (Bai et al., 2023b), enrich the long-context data from QA (NarrativeQA (Kočiskỳ et al., 2018), QuALITY (Pang et al., 2022), Qasper (Dasigi et al., 2021)) and summarization (GovReport (Huang et al., 2021), QMSum (Zhong et al., 2021)) datasets as the main components of the evaluation. Based on this, different evaluations impose varying requirements on the tasks. LEval (An et al., 2023) and CLongEval (Qiu et al., 2024) emphasize high-quality evaluation data, obtaining reliable long-context evaluation data through manual screening or annotation. M4LE (Kwan et al., 2023) highlights the diversity of data sources and categorizes long-context evaluation into five scenarios based on the distribution of answers in the text: explicit single-span, semantic single-span, explicit multiple-span, semantic multiple-span, and global context understanding. LooGLE (Li et al., 2023b) proposes evaluating long-context and short-context dependencies simultaneously. LV-Eval (Yuan et al., 2024b) focuses on QA tasks by introducing confusing facts in the context to increase the difficulty.

Long-Context Retrieval Retrieval is also a classic task in long-context evaluation, with early benchmarks such as LongEvalLi et al. (2023a) emphasizing it. Retrieval tasks offer better flexibility than QA and summarization based on naturally long texts. Needle-In-A-Haystack (NIAH) (Kamradt, 2023) is the first to propose reflecting LLM's recall performance in varying depths at varying context lengths. This sparks a surge in research on long-context retrieval tasks (Young et al., 2024; Cai et al., 2024c; Wang et al., 2024j), significantly altering the trajectory of long-context evaluation development. Notably, Gemini-1.5 (Reid et al., 2024) expands the single-NIAH to multi-NIAH, achieving impressive results. Moreover, Hsieh et al. (2024a) proposes various variants such as multikey and multivalue NIAH, creating an entirely synthetic long-context evaluation, RULER, which has become a new competitive focus among long-context LLMs (Team et al., 2024c; Liu et al., 2024d; Liquid, 2024).

Furthermore, there are also domain-specific retrievals, such as DocFinQA (Reddy et al., 2024) and Gupta et al. (2024), and structured data retrievals, namely enabling long-context LLMs to simulate SQL execution or database manipulation, including S3eval (Lei et al., 2024b), BIRD (Li et al., 2024d), Spider 2.0 (Lei et al., 2024a), HoloBench (Maekawa et al., 2024). To

Name	Time	Benchmark Feature						
		Len.	Lang.	Flexible	Stable	D.C.	Align.	L.O.
Scroll (Shaham et al., 2022)	22.01	∼8k	En	✗	✗	✗	✗	✗
ZeroScrolls (Shaham et al., 2023)	23.05	∼8k	En	✗	✗	✗	✗	✗
LEval (An et al., 2023)	23.07	4k-60k	En	✗	✓	✗	✗	✗
LongBench (Bai et al., 2023b)	23.08	∼10k	En, Zh	✗	✗	✗	✗	✗
BAMBOO (Dong et al., 2024f)	23.09	4k-16k	En	❍	✓	✓	✓	✗
M4LE (Kwan et al., 2023)	23.10	1k-128k	En, Zh	❍	✗	✗	✗	✗
LooGLE (Li et al., 2023b)	23.11	∼20k	En	❍	✗	✓	✗	✗
Marathon (Zhang et al., 2023e)	23.12	∼80k	En	✗	✓	✗	✗	✗
Needle-In-A-Haystack (Kamradt, 2023)	23.11	1k-128k	En	✓	✓	✗	✗	✗
InfiniteBench (Zhang et al., 2024q)	24.02	∼200k	En, Zh	✗	✗	✗	✗	✓
LV-Eval (Yuan et al., 2024b)	24.02	16k-56k	En	✓	✓	✓	✗	✗
Multi-NIHA (Reid et al., 2024)	24.03	1k-1M	En	✓	✓	✗	✓	✗
CLongEval (Qiu et al., 2024)	24.03	1k-100k	Zh	❍	✗	✗	✗	✗
LongICLBench (Li et al., 2024j)	24.04	2k-50k	En	✓	✓	✗	✗	✗
XL2Bench (Ni et al., 2024)	24.04	∼200k	En, Zh	✗	✗	✓	✗	✗
RULER (Hsieh et al., 2024a)	24.04	4k-1M	En	✓	✓	✗	✗	✗
Ada-LEval (Wang et al., 2024a)	24.04	2k-128k	En	❍	✓	✗	✗	✗
LoFT (Lee et al., 2024a)	24.06	32k-1M	En, Es, Fr, Hi, Zh	❍	✓	✗	✗	✗
Loong (Wang et al., 2024i)	24.06	10k-250k	En, Zh	❍	✓	✓	✗	✗
BABILong (Kuratov et al., 2024)	24.06	4k∼10M	En	✓	✓	✓	✗	✗
LongIns (Gavin et al., 2024)	24.06	256-16k	En	✓	✓	✗	✓	✗
NeedleBench (Li et al., 2024f)	24.07	20k-1M	En, Zh	✓	✓	✗	✓	✗
HelloBench (Que et al., 2024)	24.09	∼2k	En	✗	✓	✗	✓	✓
LongGenBench$_1$ (Wu et al., 2024k)	24.09	∼20k	En	❍	✓	✗	✓	✓
LongGenBench$_2$ (Liu et al., 2024n)	24.10	4k-128k	En	✓	✓	✗	✓	✓
HELMET (Yen et al., 2024b)	24.10	8k-128k	En	❍	✓	✗	✗	✗
LongSafetyBench (Huang et al., 2024a)	24.11	∼40k	En	✗	✓	✗	✓	✗
LIFBench (Wu et al., 2024i)	24.11	4k-128k	En	✓	✓	✗	✓	✗
LongBench v2 (Bai et al., 2024b)	24.12	32k-128k	En, Zh	❍	✓	✗	✗	✗
LongProc (Ye et al., 2025a)	25.01	500 8k	En	❍	✓	✗	✓	✓

Table 5: Comparison of the mainstream or comprehensive long-context benchmarks at present. The comparison includes the benchmark features such as average length, language, etc., and type of tasks including QA, summary, and retrieval in the continued table. In this table, Flexible stands for whether the length of evaluating data is flexible. Stable stands for stable evaluation. D.C. stands for data contamination. Align. stands for containing alignment tasks. L.O. stands for long output. ✓ means yes, while ✗ means no, and ❍ means the data in the benchmark are grouped into subsets by different length ranges.

improve recall accuracy and assess whether LLM truly understands the context (Gao et al., 2023; Hilgert et al., 2024; Zhang et al., 2024d), researchers also want LLM to provide relevant citations for the retrieval content (Buchmann et al., 2024; Tang et al., 2024d). Such tasks have now been integrated into emerging long-context evaluation benchmarks, such as LoFT (Lee et al., 2024a) and HELMET (Yen et al., 2024b).

Name	Type of tasks							
	QA	Summ.	Retrieval	Code	Math	Agg.	ICL	Reasoning
Scroll (Shaham et al., 2022)	✓	✓	✗	✗	✗	✗	✗	✗
ZeroScrolls (Shaham et al., 2023)	✓	✓	✗	✗	✗	✓	✗	✗
LEval (An et al., 2023)	✓	✓	✓	✓	✓	✗	✓	✗
LongBench (Bai et al., 2023b)	✓	✓	✓	✓	✗	✗	✓	✗
BAMBOO (Dong et al., 2024f)	✓	✗	✗	✓	✗	✓	✗	✗
M4LE (Kwan et al., 2023)	✓	✓	✓	✗	✗	✗	✗	✗
LooGLE (Li et al., 2023b)	✓	✓	✓	✗	✗	✓	✗	✓
Marathon (Zhang et al., 2023e)	✓	✗	✓	✗	✓	✓	✗	✓
Needle-In-A-Haystack (Kamradt, 2023)	✗	✗	✓	✗	✗	✗	✗	✗
InfiniteBench (Zhang et al., 2024q)	✓	✓	✓	✓	✓	✓	✗	✗
LV-Eval (Yuan et al., 2024b)	✓	✗	✓	✗	✗	✗	✗	✗
Multi-NIHA (Reid et al., 2024)	✗	✗	✓	✗	✗	✗	✗	✗
CLongEval (Qiu et al., 2024)	✓	✓	✓	✗	✗	✗	✗	✗
LongICLBench (Li et al., 2024j)	✓	✗	✗	✗	✗	✗	✓	✗
XL2Bench (Ni et al., 2024)	✓	✓	✓	✗	✗	✗	✗	✗
RULER (Hsieh et al., 2024a)	✓	✗	✓	✗	✗	✓	✗	✗
Ada-LEval (Wang et al., 2024a)	✗	✗	✗	✗	✗	✓	✗	✗
LoFT (Lee et al., 2024a)	✓	✗	✓	✓	✗	✗	✓	✗
Loong (Wang et al., 2024i)	✓	✗	✓	✗	✗	✓	✗	✓
BABILong (Kuratov et al., 2024)	✓	✗	✓	✗	✗	✗	✗	✓
LongIns (Gavin et al., 2024)	✓	✗	✓	✗	✗	✗	✗	✗
NeedleBench (Li et al., 2024f)	✓	✗	✓	✗	✗	✗	✗	✓
HelloBench (Que et al., 2024)	✓	✓	✗	✗	✗	✗	✗	✗
LongGenBench_1 (Wu et al., 2024k)	✓	✗	✗	✗	✓	✗	✗	✓
LongGenBench_2 (Liu et al., 2024n)	✓	✗	✗	✗	✓	✗	✗	✗
HELMET (Yen et al., 2024b)	✓	✓	✓	✗	✗	✓	✓	✗
LongSafetyBench (Huang et al., 2024a)	✓	✗	✓	✗	✗	✗	✓	✗
LIFBench (Wu et al., 2024i)	✓	✗	✓	✗	✗	✓	✗	✗
LongBench v2 (Bai et al., 2024b)	✓	✗	✗	✓	✗	✗	✓	✓
LongProc (Ye et al., 2025a)	✓	✗	✗	✓	✓	✓	✗	✓

Table 6: The continued table of Table 5 comparing the type of tasks in the mainstream or comprehensive long-context benchmarks at present. QA stands for question-answer tasks. Summ. stands for summarization tasks. Retrieval stands for retrieval task. Code stands for code tasks. Math stands for math tasks. Agg. stands for aggregation tasks. ICL stands for long in-context learning tasks. Reasoning stands for reasoning tasks.

Due to the popularity of retrieval tasks, discussions on retrieval have also emerged. For example, Liu et al. (2024l) and An et al. (2023) highlight that LLMs tend to recall topics at the beginning and end of a context more easily and make mistakes with topics in the middle, thus demonstrating the Lost-In-the-Middle phenomenon. Furthermore, Koo et al. (2024) separates QA and evidence selection within retrieval from the perspective of task alignment. Yu et al. (2024c) divides retrieval into matching and logical retrieval, exploring the corresponding improving methods. Goldman et al. (2024) analyzes long-context evaluation from the recall perspective and proposes two orthogonal dimensions, dispersion, and scope, to identify potential directions for more challenging long-context evaluations.

Code, Math, and Aggregation In addition to tasks focusing on long natural language text, there are also long-context tasks centered on logical languages such as code and mathematics. Regarding code, LEval (An et al., 2023) and LongBench (Bai et al., 2023b) are the first to incorporate code into long-context evaluation, which has been inherited by subsequent benchmarks (Zhang et al., 2024q; Bai et al., 2024b). Additionally, there are tasks specifically aimed at repository-level long code, such as RepoQA (Liu et al., 2024k). Regarding math, LEval (An et al., 2023) and LongGenBench$_2$ (Liu et al., 2024n) expand the short-context task GSM8k (Cobbe et al., 2021) into a long-context task using many-shot ICL and question concatenation respectively. In contrast, Marathon (Zhang et al., 2023e) and InfiniteBench (Zhang et al., 2024q) introduced more complex long-context mathematical computation tasks, while LongGenBench$_1$ (Wu et al., 2024k) examined the LLM's spatial-temporal understanding in long context.

Besides, there is also a category of long-context evaluation that includes sorting and statistics, generally referred to as aggregation tasks (Shaham et al., 2023; Hsieh et al., 2024a). Aggregation tasks are first mentioned in LRA (Tay et al., 2021) and introduced into text evaluation in ZeroScrolls (Shaham et al., 2023), which includes positive review statistics and summary sorting. After that, sorting tasks still exists in (Dong et al., 2024f; Li et al., 2023b; Zhang et al., 2023e; 2024q; Wang et al., 2024a), while the recent HELMET evaluation suite also includes re-ranking tasks (Yen et al., 2024b). Regarding statistics, finding the maximum number and identifying (Zhang et al., 2024q) the most frequent words (Hsieh et al., 2024a) are also proposed. Although aggregation tasks often occur in long-context benchmarks, they are less emphasized due to the deviation from natural long texts (Hsieh et al., 2024a).

Long In-Context Learning Regarding LLMs, the two most notable capabilities are ICL (Brown et al., 2020; Pan et al., 2023) and reasoning (Wei et al., 2022), and long-context provides a deeper exploration of both. For ICL, a longer context enables more demonstrations, offering greater potential to stimulate LLM. Long ICL is first introduced in long-context evaluation in LEval (An et al., 2023) and LongBench (Bai et al., 2023b), primarily to extend the context of short-context tasks. After Gemini-1.5 (Reid et al., 2024) prompts LLM to learn new languages with the grammar book and dictionary, long ICL has become a new focus for long-context evaluation (Li et al., 2024j; Agarwal et al., 2024).

For example, LongICLBench (Li et al., 2024j) evaluates a wide range of long-context LLMs and finds that most can benefit from extensive demonstrations when the length is within a certain range. As the input grows longer, it will lead to a performance fluctuation or decline (Li et al., 2024j). Bertsch et al. (2024) further points out that long ICL is sensitive to the distribution of demonstrations, and when there are enough demonstrations, the effect of the sampling method diminishes. Other studies indicate that long ICL is also influenced by the quality of the demonstrations (Li et al., 2024o; Agarwal et al., 2024), precision (Wang et al., 2024e), retrieval (Zou et al., 2024b), and other factors (Agarwal et al., 2024). Additionally, Wang et al. (2024c) propose General Purpose In-Context Learning, covering more domains including decision-making and world modeling through continuous generation and interaction. Long ICL has become a significant sub-item in emerging long document evaluation standards, such as LoFT (Lee et al., 2024a) and HELMET (Yen et al., 2024b), and further discussions on long ICL will be present in **Q10** in Section12.

Long-Context Reasoning The discussion of long reasoning can be traced back to early multi-hop reasoning tasks, such as HotpotQA (Yang et al., 2018) and MuSiQue (Trivedi et al., 2022). The emergence of long context provides more exploration space for multi-hop reasoning. For example, Variable Tracing in RULER, CountingStars (Song et al., 2024b), Loong (Wang et al., 2024i), BABILong (Kuratov et al., 2024), and Needlebench (Li et al., 2024f) ask models to aggregate multi-hop evidence inserted in the long-context when answering the final question. However, these evaluations still tend to focus on synthetic texts, lacking assessments of reasoning capabilities in real-world scenarios.

Apart from explicit multi-hop reasoning, some benchmarks (Li et al., 2023b; Zhang et al., 2023e; Bai et al., 2024b), also regard a deeper understanding of context as a type of reasoning. Recently, NovelQA (Wang et al., 2024b), NoCha (Karpinska et al., 2024), and DetectiveQA (Xu et al., 2024f) design reasoning evaluations for native long texts, leveraging the

complex reasoning chains present in long novels. Moreover, NovelQA and DetectiveQA require LLM to output its reasoning process and conduct a process-centered evaluation (Wang et al., 2024b; Xu et al., 2024f), which offers a more realistic and challenging evaluation. More discussion on long-context reasoning and long output will be shown **Q9** in Section12.

In addition to the aforementioned evaluation tasks, there are long-context evaluations on other traditional NLP tasks. For example, some tasks in M4LE (Kwan et al., 2023) involve text classification. StNLab in CLongEval (Qiu et al., 2024) explores the annotation issues in long Chinese texts. Manikantan et al. (2024) and Vodrahalli et al. (2024) focuses on referential understanding within long texts.

11.2 Benchmark Features

Length Length is an important feature for long-context evaluations. Before retrieval tasks like NIAH (Kamradt, 2023; LangChain, 2024) mark a turning point, the length of long-context evaluation benchmarks lags behind the lengths reported by long-context LLMs (Peng et al., 2024b; Young et al., 2024; Cai et al., 2024c). This is primarily due to the limited native long-context corpora, making it difficult to enrich long-context evaluation (An et al., 2023; Li et al., 2023b). After this point, the situation reverses. On one hand, the length of synthetic tasks is flexible, and any length for evaluation is allowed (Liu et al., 2024o; Lieber et al., 2024). On the other hand, researchers begin proposing more challenging evaluations (Levy et al., 2024; Li et al., 2024j; Hsieh et al., 2024a; Gavin et al., 2024), discovering that long-context LLMs fail to achieve acceptable performance within the claimed context lengths.

In addition to length itself, flexibility is a key feature of long-context evaluations (Kamradt, 2023; Yen et al., 2024b). As mentioned earlier, traditional long-context benchmarks (An et al., 2023; Bai et al., 2023b; Zhang et al., 2024q) are not scalable and only able to measure performance at different context lengths by truncating to various lengths (Bai et al., 2023b). Subsequent synthetic task evaluations (Kamradt, 2023; Levy et al., 2024; Hsieh et al., 2024a; Liu et al., 2024n), generally allow for customized context length. Additionally, some evaluation benchmarks (Kwan et al., 2023; Lee et al., 2024a; Yen et al., 2024b) group the evaluation data to different subsets by different length ranges, representing a trade-off.

Stability Another important feature for long-context evaluation is stability, a persistent pain point in long-context evaluation (Novikova et al., 2017; An et al., 2023; Yen et al., 2024b). Specifically, it is difficult to provide a reliable evaluation metric for generative tasks such as long QA, summarization, and open-ended generation (An et al., 2023; Novikova et al., 2017) which are common in long-context benchmarks. In response, different long-context evaluation benchmarks have proposed various solutions. First, benchmarks like Dong et al. (2024f) avoid this issue by directly discarding long output tasks. Next, benchmarks like Zhang et al. (2023e); Lee et al. (2024a); Bai et al. (2024b) address the problem by transforming generative answers into multiple-choice questions or constraining evaluation metrics.

Furthermore, some long-context research has delved more deeply into the stability of long output evaluations. LEval (An et al., 2023) is the first to propose using LLMs to compute reference-free, pairwise win rates to measure the quality of long outputs. After that, LV-EvalYuan et al. (2024b) improves the stability of output measurement through a keyword-recall-based metric design without the aid of LLMs. In contrast, DetectiveQA (Xu et al., 2024f) introduced a step-wise reasoning metric that compares the reasoning chains of the model outputs to reference steps, measuring long reasoning based on the recall of reasoning steps. Similarly, HELMET (Yen et al., 2024b) breaks down conventional summarization references into atomic claims and then has LLMs evaluate their recall. HelloBench (Que et al., 2024), based on the ordinary LLM-as-a-Judge (Zheng et al., 2023), decomposes answer quality into a linear combination of multiple scoring items from LLMs in a checklist, thereby reducing bias in LLM judges. Additionally, there are other solutions involving task formats, such as ProxyQA (Tan et al., 2024a), which evaluates a model's performance based on the outputs of the model under meta-questions to reflect the long-text generation capability of the model being tested, as well as Liu et al. (2024n).

Data Contamination Evaluation benchmarks always face the issue of data contamination (Golchin & Surdeanu, 2024), and avoiding it is an important topic. In response, BAMBOO (Dong et al., 2024f) and LooGLE (Li et al., 2023b) are the first to propose constructing evaluation sets using newly crawled data to mitigate this problem. Besides, LV-Eval (Yuan et al., 2024b) and XL2Bench (Ni et al., 2024) employed keyword, phrase, and text replacement methods to address the issue. Additionally, DetectiveQA (Xu et al., 2024f) suggests comparing model performance under context-free scenarios to determine whether LLM relies on internal knowledge rather than context to answer questions. Finally, some studies (Wang et al., 2024i; Kuratov et al., 2024) claim that the data contamination may not exist for particularly long or very general texts.

Alignment Evaluation Finally, some long-text evaluations also discuss the alignment performance of long-context LLMs. On one hand, LongIns (Gavin et al., 2024) and LIFBench (Wu et al., 2024i) examine the instruction-following performance of long-context LLMs, with LongInsc(Gavin et al., 2024) reporting that the effective context length for instruction following is significantly shorter than the claimed context length of long-context LLMs. On the other hand, Many-shot Jailbreaking (Anil et al., 2024) focuses on the long-context safety performance of long-context LLMs, finding that numerous demonstrations can disrupt model alignment under long-context attacks. Subsequently, Huang et al. (2024a) and Roberts et al. (2024) offer a broader discussion of long-context safety, exploring safety issues in various scenarios. Besides, some long-context evaluations investigate the memory capabilities of LLMs in real-world interactions (Thonet et al., 2024; Hosseini et al., 2024b).

Additionally, there are some domain-specific long-context benchmarks such as Hosseini et al. (2024a) in the medical domain and Reddy et al. (2024) in the financial domain.

12 Unanswered Questions

In the 10 sections above, we have illustrated the development trajectory of long-context from the extensive literature in different aspects. In this section, we can make a more comprehensive conclusion in the final section, unlike the previous survey focused on particular domains (Huang et al., 2023; Zhao et al., 2023b; Pawar et al., 2024; Luohe et al., 2024). However, instead of listing some take-home messages or definitive claims, inspired by the masterpiece of Richard Strauss, we are more willing to end our journey with a longer context with 10 unanswered questions, to stimulate more in-depth thoughts and research on long-context LLMs from these perspectives. Whatever the answers may be, we believe we come out of reading this survey, wiser and better people than before.

Q1 Position Bias While considerable efforts have been devoted to augmenting the context window length of LLMs (Chen et al., 2023b; bloc97, 2023a; Peng et al., 2024b), position bias persists within these models. Position bias refers to LLMs' propensity to favor certain positions over others (Wang et al., 2023d; Zheng et al., 2023). A notable manifestation of this bias is the phenomenon known as ***lost in the middle***, where LLMs tend to allocate anomalously higher attention to the beginning and end of context, while the middle part receives relatively less focus (Liu et al., 2024l). This tendency is further exacerbated by what has been termed the ***attention sink*** effect, wherein the majority of attention scores are concentrated on the initial tokens of the context (Xiao et al., 2024c). Surprisingly, such bias is observed even in NoPE-based LLM, where no position information is explicitly injected, but the performance of NIAH still declines from the middle (Wang et al., 2024g). On one hand, this bias has benefited research in streaming processing (Xiao et al., 2024c; Yang et al., 2024h) and KV cache optimization (Tang et al., 2024a; Xiao et al., 2024b). On the other hand, many empirical efforts have also been devoted to addressing this bias (Zhang et al., 2024h; McIlroy-Young et al., 2024; Hsieh et al., 2024b), such as fill-in-the-middle (An et al., 2024c). However, minor studies try to answer why this bias exist (Gu et al., 2024b). The theoretical understanding of related mechanisms is still an unanswered question.

In parallel, Levy et al. (2024) examines the impact of input length on the inference performance of LLMs, observing a significant performance decline though the input length is still shorter than the maximum context length. Leveraging this effect, some evaluation

datasets have increased the length of evaluation texts to obscure relevant information and enhance the evaluation difficulty (Yuan et al., 2024b; Hsieh et al., 2024a; Li et al., 2024j). However, questions regarding this aspect remain relatively unsolved. Though we can easily extrapolate the LLMs to a longer context, we often struggle to guarantee an exhaustive short-to-long generalization in downstream tasks (Li et al., 2024j; Anil et al., 2024).

Q2 RoPE Design RoPE(Su et al., 2024) has emerged as the mainstream position embedding for LLMs due to its superior performance(Dubey et al., 2024; Bai et al., 2023a; Liu et al., 2024a). However, regarding length extrapolation, the current RoPE scaling methods (Roziere et al., 2023; Xiong et al., 2024a), can only achieve weak extrapolation for an infinite context length or strong extrapolation for a finite one. In strong extrapolation, RoPE-based LLMs rely on the high-dimensional, low-frequency features to represent long-context dependencies at greater distances (Barbero et al., 2024; Hong et al., 2024; Zhong et al., 2024a). However, these dimensions present OOD in extrapolation. Besides, even when position information is not OOD, the increased attention entropy also harms the long-context performance (Peng et al., 2024b; Han et al., 2024). The conflicts between periodicity and monotonicity and between full attention and attention entropy are the inherent drawbacks of scaling RoPE-based LLMs to an infinite context (Liu et al., 2024p; Men et al., 2024; Han et al., 2024).

Given these limitations of RoPE, researchers have explored additional approaches based on alternative position embedding design (Kazemnejad et al., 2024; Wang et al., 2024g), or cache operation (Xiao et al., 2024a; Liu et al., 2024o), to address these challenges. Regarding RoPE itself, the modification for length extrapolation also simulates modifications for other perspectives, such as the selection of rotary angles (Wu et al., 2024j), the number of dimensions for RoPE (GLM et al., 2024; Biderman et al., 2023), the index schema for RoPE (Golovneva et al., 2024), whether there are better design alternatives for RoPE (Sun et al., 2022b; Chi et al., 2022), how the scaling laws change under these alternative designs, and even how to design RoPE for multi-modal information (Su, 2024; Wang et al., 2024j; Li et al., 2024g), all remain open questions await deeper investigation.

Q3 Dilemma of Perplexity For a long time, perplexity has been a primary indicator for determining the upper bound of the length extrapolation (Press et al., 2022; Liu et al., 2024p). However, subsequent research has found that perplexity does not truly reflect the performance of LLMs in the downstream tasks of long context (Men et al., 2024; Hu et al., 2024h; Fang et al., 2024b; Gao et al., 2024d; Xiao et al., 2024c). Despite this, there are still works that define long-context quality based on perplexity, such as LongWanjuan (Lv et al., 2024a) and ProLong (Chen et al., 2024c) with perplexity-based metrics to compare the information gain of long contexts with short ones. Recently, LongPPL (Fang et al., 2024b) based on the comparison between sliding window perplexity and standard perplexity, is proposed to reflect LLM's real downstream performance more accurately.

However, both definitions of short-context perplexity have flaws: chunking breaks long-context dependencies while sliding windows imply that the receptive field increases with model depth. Additionally, the perplexity of different LLMs may vary due to differences in their training data distributions. Therefore, there is much space for improving perplexity in assessing LLMs performance and data quality in long-context scenarios.

Q4 Long Context v.s. RAG The choice between long-context LLMs and RAG has been a topic of debate. Xu et al. (2023) suggests that retrieval-augmented approaches allow LLMs with smaller context windows to perform on par with larger context window LLMs, and even improve the performance of long-context LLMs. However, Li et al. (2024r) has reached the opposite conclusion, indicating that under their experimental setup, long-context LLMs generally outperforms RAG. Moreover, Leng et al. (2024) indicates that using longer context does not uniformly increase RAG performance while Jiang et al. (2024c) holds an opposite opinion. Li et al. (2024l) conducts a more in-depth investigation into this issue. This raises two intriguing question: which paradigm represents the better approach for generation? Should these two paradigm be combined?

To begin with, long-context LLMs offer more complete contextual information compared to RAG, but they also come with challenges such as lower information density and high

computational resource consumption. KV cache is position-sensitive while RAG is position-independent. Whether the positional relationships within long-context LLMs play a significant role in generation remains an important topic for exploration (Bertsch et al., 2024). In contrast, RAG is more lightweight and better suited for edge devices, but it is unable to handle special scenarios, such as long outputs. Many attention acceleration or approximation methods utilize retrieval (Zhang et al., 2023f; Li et al., 2024n), raising the question of whether long-context generation can be unified with RAG. We believe that a more flexible memory-based approach, which can leverage both text and KV cache, may represent a promising and potentially superior generation paradigm in the future (Yang et al., 2024c).

Q5 Discussion on New Architecture Recent advances in LLM's architectures have revealed an intriguing pattern: the incorporation of local interaction, such as the token shift in RWKV (Peng et al., 2023a; 2024a) or convolution in Mamba (Gu & Dao, 2023). While these architectural choices designed for a long context are different, they coincidentally introduce similar mechanisms to modeling local interaction. This raises the question of whether traditional RNN, LSTM, or SSM can achieve long-context capabilities comparable to Transformer by incorporating local interaction mechanisms. Furthermore, does the standard self-attention mechanism equal the combination of local interaction based on CNN or token shift and long-context dependency captured with RNN or SSM, and why or why not?

A potential explanation lies in the mechanisms of information processing. In attention-based architecture, information from different positions is processed in parallel before fusion (Vaswani et al., 2017), whereas RNN, LSTM, and SSM architectures process information from various distances (both short and long-range) simultaneously (Beck et al., 2024; Gu & Dao, 2023). This mixed processing can lead to mutual interference, where short-context information may disrupt the modeling of long-context dependencies, and vice versa. The introduction of convolution or token shift represents an attempt to decouple information interaction across different scales. Moreover, such assumptions also need further validation.

Q6 On-Device Long Context The future of long-context LLMs also involves edge-based multi-modal applications, which require locally deployed models as a foundation or important support. Although major AI companies are integrating their models into local software (Wu et al., 2024d; Yin et al., 2024b), these solutions still rely on LLMs in the cloud. The future interaction paradigm will be fundamentally multi-modal (Yao et al., 2024e), processing and generating across multiple modalities such as speech, images, text, and action sequences (Google, 2024b; Barkley, 2024; Apple, 2024; Google, 2024a). Meanwhile, to reduce latency, ensure privacy, balance server loads, and enable personalization, a substantial portion of computation and storage tasks of long-context LLMs will migrate closer to users, specifically to edge devices (Wu et al., 2024d; Xu et al., 2024a).

Although the direction of development is clear, many challenges remain in delivering a smooth and natural long-context interaction experience to users. These challenges span multiple domains (Xu et al., 2024a): How can algorithms, hardware, and software be further optimized to reduce the resource footprint of long-context operations and improve inference speed (MLC team, 2023-2024; Lu et al., 2024b; Xue et al., 2024c; Choe et al., 2024)? What technologies are needed for more seamless integration (Yao et al., 2024e; Yin et al., 2024b)? Is it possible to achieve horizontal scaling of long-context LLMs in this process and make it close to users? These challenges await researchers and engineers to solve them. Since the integration of large language models and edge devices has already become an industry-wide consensus (Qualcomm, 2023; Apple, 2024; Qualcomm, 2024; Google, 2024a; Lu et al., 2024b), we hope that they will all be resolved in the near future.

Q7 Long-Context Training from Scratch From a perspective of model capability, training with long-context data from the start offers several advantages. It naturally enhances LLM's ability to handle longer context (Gao et al., 2024d). Following the "the best part is no part" philosophy, it eliminates the need for complex length adaptation techniques. Training with mixed-length texts in the same batch allows LLMs to learn from text length distributions that better reflect real-world scenarios (Gao et al., 2024d; ChatGLM, 2024).

The challenges of training with mixed-length sequences in the same batch are primarily engineering-related rather than theoretical. Traditional training frameworks require extensive padding when processing texts of varying lengths, which wastes computational resources and reduces training throughput. The disparity in computational load between long and short texts creates load imbalance issues in distributed training environments. Sophisticated runtime dynamic schedulers may be needed to address these challenges. Therefore, improving long-context training efficiency remains a critical engineering challenge, with a particular focus on enhancing the efficiency of mixed-length text training.

Q8 Quantity and Quality of Long Data As reported by Ilya, existing corpora have almost been exhausted for pre-training[3]. The scarcity of data is more severe for long context (ChatGLM, 2024; Gao et al., 2024b). Although researchers have constructed longer textual data by fancy concatenation (Shi et al., 2024; ChatGLM, 2024; Zhao et al., 2024b) or task-oriented synthesis (An et al., 2024c; Pham et al., 2024), concerns about the effectiveness of synthetic data have never ceased (Gao et al., 2024b; Zhao et al., 2024e; Que et al., 2024).

Besides quantity, quality also matters. Unfortunately, the definition of long-context data quality has not been thoroughly explored (Lv et al., 2024a; Chen et al., 2024c), and more researchers are trying to optimize the quality of data mixing between long and short corpora (Xiong et al., 2024a; ChatGLM, 2024; Gao et al., 2024d). However, training with limited long texts fails to guarantee the short-to-long generalization. (Levy et al., 2024; Hsieh et al., 2024a; Anil et al., 2024; Huang et al., 2024a), and whether training on longer texts will harm short-context performance is also a question (An et al., 2024b; Gao et al., 2024d).

In the multi-modal domain, the scarcity of long video is also significant (Qian et al., 2024a; Yin et al., 2024a; Ren et al., 2024b). Moreover, although both text and video share sequential features, the generalization from long-context text to long-context video, and from long-context reasoning in text to long-context reasoning in video (Li et al., 2024e), remains a topic that requires further research and discussion.

Q9 Long Output and Reasoning In the last two questions, we will finally discuss how to enhance the model capabilities with long context. Long context involves long input and long output and we start with the latter. Compared to short outputs, long outputs involve more complex dependencies and exposure bias resulting from inconsistencies between the previously generated content and the ongoing output (An et al., 2022). These factors make training long-output LLMs particularly challenging. Although some evaluation work on long-text outputs has been conducted (Tan et al., 2024a; Que et al., 2024), there is still a lack of effective metrics for assessing long outputs. Manual scoring remains subjective and difficult, and LLM-as-a-Judge requires further in-depth exploration (Dubois et al., 2024; Que et al., 2024). Thus, evaluating long-output LLMs remains a significant challenge.

Furthermore, the expectations for LLM outputs go beyond mere content generation. There is a need for LLMs to solve complex reasoning problems. The rapid rise of o1 has also highlighted the tremendous potential of long reasoning (OpenAI, 2024; Zeng et al., 2024a), and long output, as a core capability of long reasoning, needs to achieve better performance to support the advancement of related work. Moreover, Snell et al. (2024) indicates that scaling at test-time is crucial, and long output is a promising method to achieve it. Despite the vast application potential of long-output generation, it still faces numerous challenges. For instance, maintaining logical and informational consistency during long-text generation, and effectively controlling aspects such as style, tone, and emotion in the generated content, remain key issues for researchers (Bai et al., 2024c; Quan et al., 2024). Additionally, these questions are more severe in MLLMs for multi-modal dependencies (Tan & Bansal, 2019; Zhang et al., 2021) and cross-modal consistency (Zhang et al., 2024o; Chou et al., 2024).

Q10 Long In-Context Learning and More Long In-Context Learning is a method for enhancing LLM performance through long inputs (Brown et al., 2020; Pan et al., 2023; Agarwal et al., 2024). Current discussions on long ICL mainly focus on benchmarks that use

[3]Ilya Sutskever's talk at NeurIPS 2024. Sequence to Sequence Learning with Neural Networks. https://www.youtube.com/watch?v=qo-ZjF_LAz8

it to analyze long-context capabilities of LLMs (Li et al., 2024j; Wang et al., 2024c). However, there is still a lack of attempts to treat long in-context learning as a means to overcome LLM limitations through long contexts (Bertsch et al., 2024; Agarwal et al., 2024). Although some LLMs (Reid et al., 2024) have been shown to achieve translations of a brand new language using its grammar book and numerous demonstrations, discussions on related technical roadmap remain limited and require more open-source reproductions. Some studies also attempt to establish scaling or interpreting mechanisms between long ICL and SFT (Dai et al., 2023; Mosbach et al., 2023), but certain theoretical analyses still need to be based on the separability assumption of softmax operations (Dai et al., 2023), leaving a gap in practice.

Furthermore, some works also explore test-time training (Sun et al., 2020; 2024f), the idea of training certain parameters of LLMs through a long context to enhance LLM capabilities or perceive user preferences. However, the context lengths involved in these works are still not sufficiently long and similarly lack corresponding scaling mechanisms. In the research on long outputs, concepts such as test-time scaling (Snell et al., 2024; OpenAI, 2024; Zeng et al., 2024a) have emerged to enhance LLM performance by increasing the computational overhead of inference. However, the source of computational overhead, from long inputs or long outputs, is not clearly defined. Whether scaling inputs or scaling outputs yields more benefits is also a topic for discussion. Finally, these learning paradigms represent attempts to treat LLMs as humans, striving for ultimate life-long learning. This process will also compel us to rethink the architecture, infrastructure, and training strategy to suit these training paradigms, allowing LLMs to learn in the interactions until the Ewigkeit.

References

Index1.9b technical report. 2024.

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024a.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024b.

Shantanu Acharya, Fei Jia, and Boris Ginsburg. Star attention: Efficient llm inference over long sequences. *arXiv preprint arXiv:2411.17116*, 2024.

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Luis Rosias, Stephanie CY Chan, Biao Zhang, Aleksandra Faust, and Hugo Larochelle. Many-shot in-context learning. In *ICML 2024 Workshop on In-Context Learning*, 2024.

Amey Agrawal, Nitin Kedia, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav Gulavani, Alexey Tumanov, and Ramachandran Ramjee. Taming {Throughput-Latency} tradeoff in {LLM} inference with {Sarathi-Serve}. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 117–134, 2024.

Meta AI. Llama 3.3 - 70b instruct, 2024. URL https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. Etc: Encoding long and structured inputs in transformers. *arXiv preprint arXiv:2004.08483*, 2020.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

Yash Akhauri, Safeen Huda, and Mohamed S Abdelfattah. Attamba: Attending to multi-token states. *arXiv preprint arXiv:2411.17685*, 2024.

Keivan Alizadeh, Iman Mirzadeh, Dmitry Belenko, Karen Khatamifard, Minsik Cho, Carlo C Del Mundo, Mohammad Rastegari, and Mehrdad Farajtabar. Llm in a flash: Efficient large language model inference with limited memory. *arXiv preprint arXiv:2312.11514*, 2023.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

Reza Yazdani Aminabadi, Samyam Rajbhandari, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Olatunji Ruwase, Shaden Smith, Minjia Zhang, Jeff Rasley, et al. Deepspeed-inference: enabling efficient inference of transformer models at unprecedented scale. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–15. IEEE, 2022.

Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. Cont: Contrastive neural text generation. *Advances in Neural Information Processing Systems*, 35: 2197–2210, 2022.

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. L-eval: Instituting standardized evaluation for long context language models. *arXiv preprint arXiv:2307.11088*, 2023.

Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. Training-free long-context scaling of large language models. *arXiv preprint arXiv:2402.17463*, 2024a.

Chenxin An, Jun Zhang, Ming Zhong, Lei Li, Shansan Gong, Yao Luo, Jingjing Xu, and Lingpeng Kong. Why does the effective context length of llms fall short? *arXiv preprint arXiv:2410.18745*, 2024b.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, and Jian-Guang Lou. Make your llm fully utilize the context. *arXiv preprint arXiv:2404.16811*, 2024c.

Sotiris Anagnostidis, Dario Pavllo, Luca Biggio, Lorenzo Noci, Aurelien Lucchi, and Thomas Hofmann. Dynamic context pruning for efficient and interpretable autoregressive transformers. *Advances in Neural Information Processing Systems*, 36, 2024.

Cem Anil, Esin Durmus, Nina Rimsky, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel J Ford, et al. Many-shot jailbreaking. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Anonymous. LongPO: Long context self-evolution of large language models through short-to-long preference optimization. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=qTrEq31Shm. under review.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, pp. 929–947, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703850. doi: 10.1145/3620665.3640366. URL https://doi.org/10.1145/3620665.3640366.

Anthropic. Introducing claude, 2023. URL https://www.anthropic.com/news/introducing-claude.

Anthropic. Model card and evaluations for claude models, 2024a. URL https://www-cdn.anthropic.com/bd2a28d2535bfb0494cc8e2a3bf135d2e7523226/Model-Card-Claude-2.pdf.

Anthropic. Introducing the next generation of claude, 2024b. URL https://www.anthropic.com/news/claude-3-family.

Apple. Apple intelligence is available today on iphone, ipad, and mac, 2024. URL https://www.apple.com/hk/en/newsroom/2024/10/apple-intelligence-is-available-today-on-iphone-ipad-and-mac/.

Daiyaan Arfeen, Zhen Zhang, Xinwei Fu, Gregory R. Ganger, and Yida Wang. Pipefill: Using gpus during bubbles in pipeline-parallel llm training. *arXiv preprint arXiv:2410.07192*, 2024. URL https://arxiv.org/abs/2410.07192.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023a.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*, 2023b.

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. Longalign: A recipe for long context alignment of large language models. *arXiv preprint arXiv:2401.18058*, 2024a.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024b.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longwriter: Unleashing 10,000+ word generation from long context llms. *arXiv preprint arXiv:2408.07055*, 2024c.

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Veličković. Round and round we go! what makes rotary positional encodings useful? *arXiv preprint arXiv:2410.06205*, 2024.

Warren Barkley. The prompt: What is long context — and why does it matter for your ai?, 2024. URL https://cloud.google.com/transform/the-prompt-what-are-long-context-windows-and-why-do-they-matter.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory. *arXiv preprint arXiv:2405.04517*, 2024.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. Decimamba: Exploring the length extrapolation potential of mamba. *arXiv preprint arXiv:2406.14528*, 2024.

Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*, 2024.

Zhengda Bian, Hongxin Liu, Boxiang Wang, Haichen Huang, Yongbin Li, Chuan-Qing Wang, Fan Cui, and Yang You. Colossal-ai: A unified deep learning system for large-scale parallel training. *Proceedings of the 52nd International Conference on Parallel Processing*, 2021. URL https://api.semanticscholar.org/CorpusID:240070340.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Tsachi Blau, Sharon Fogel, Roi Ronen, Alona Golts, Roy Ganz, Elad Ben Avraham, Aviad Aberdam, Shahar Tsiper, and Ron Litman. Gram: Global reasoning for multi-page vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15598–15607, 2024.

bloc97. Dynamically scaled rope further increases performance of long context llama with zero fine-tuning, July 2023a. URL https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/.

bloc97. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation., June 2023b. URL https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/.

William Brandon, Aniruddha Nrusimha, Kevin Qian, Zack Ankner, Tian Jin, Zhiye Song, and Jonathan Ragan-Kelley. Striped attention: Faster ring attention for causal transformers. *arXiv preprint arXiv:2311.09431*, 2023.

William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan Kelly. Reducing transformer key-value cache size with cross-layer attention. *arXiv preprint arXiv:2405.12981*, 2024.

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

Jan Buchmann, Xiao Liu, and Iryna Gurevych. Attribute or abstain: Large language models as long document assistants. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8113–8140, 2024.

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35:11079–11091, 2022.

Aydar Bulatov, Yuri Kuratov, Yermek Kapushev, and Mikhail S Burtsev. Scaling transformer to 1m tokens and beyond with rmt. *arXiv preprint arXiv:2304.11062*, 2023.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. *arXiv preprint arXiv:2405.17430*, 2024a.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024b.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang

Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. Internlm2 technical report, 2024c.

Zouying Cao, Yifei Yang, and Hai Zhao. Head-wise shareable attention for large language models. *arXiv preprint arXiv:2402.11819*, 2024.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S Abdelfattah, and Kai-Chiang Wu. Palu: Compressing kv-cache with low-rank projection. *arXiv preprint arXiv:2407.21118*, 2024.

ChatGLM. Glm long scaling: Pre-trained model contexts to millions, 2024. URL https://medium.com/@ChatGLM/glm-long-scaling-pre-trained-model-contexts-to-millions-caa3c48dea85. Accessed: 2024-12-25.

Guanzheng Chen, Xin Li, Zaiqiao Meng, Shangsong Liang, and Lidong Bing. Clex: Continuous length extrapolation for large language models. In *The Twelfth International Conference on Learning Representations*, 2024a.

Guo Chen, Yifei Huang, Jilan Xu, Baoqi Pei, Zhe Chen, Zhiqi Li, Jiahao Wang, Kunchang Li, Tong Lu, and Limin Wang. Video mamba suite: State space model as a versatile alternative for video understanding. *arXiv preprint arXiv:2403.09626*, 2024b.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*, 2023a.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2025.

Longze Chen, Ziqiang Liu, Wanwei He, Yunshui Li, Run Luo, and Min Yang. Long context is not long at all: A prospector of long-dependency data for large language models. *arXiv preprint arXiv:2405.17915*, 2024c.

Ping Chen, Wenjie Zhang, Shuibing He, Yingjie Gu, Zhuwei Peng, Kexin Huang, Xuan Zhan, Weijian Chen, Yi Zheng, Zhefeng Wang, et al. Optimizing large model training through overlapped activation recomputation. *arXiv preprint arXiv:2406.08756*, 2024d.

Qiaoling Chen, Diandian Gu, Guoteng Wang, Xun Chen, YingTong Xiong, Ting Huang, Qinghao Hu, Xin Jin, Yonggang Wen, Tianwei Zhang, et al. Internevo: Efficient long-sequence large language model training via hybrid parallelism and redundant sharding. *arXiv preprint arXiv:2401.09149*, 2024e.

Qiaoling Chen, Qinghao Hu, Guoteng Wang, Yingtong Xiong, Ting Huang, Xun Chen, Yang Gao, Hang Yan, Yonggang Wen, Tianwei Zhang, et al. Lins: Reducing communication overhead of zero for efficient llm training. In *2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, pp. 1–10. IEEE, 2024f.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018a.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023b.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. {TVM}: An automated {End-to-End} optimizing compiler for deep learning. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pp. 578–594, 2018b.

Yilong Chen, Linhao Zhang, Junyuan Shang, Zhenyu Zhang, Tingwen Liu, Shuohuan Wang, and Yu Sun. Dha: Learning decoupled-head attention from transformer checkpoints via adaptive heads fusion. *arXiv preprint arXiv:2406.06567*, 2024g.

Yingfa Chen, Xinrong Zhang, Shengding Hu, Xu Han, Zhiyuan Liu, and Maosong Sun. Stuffed mamba: State collapse and state capacity of rnn-based long-context modeling. *arXiv preprint arXiv:2410.07145*, 2024h.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. In *The Twelfth International Conference on Learning Representations*, 2024i.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024j.

Zhi Chen, Qiguang Chen, Libo Qin, Qipeng Guo, Haijun Lv, Yicheng Zou, Wanxiang Che, Hang Yan, Kai Chen, and Dahua Lin. What are the essential factors in crafting effective long context multi-hop instruction datasets? insights and best practices. *arXiv preprint arXiv:2409.01893*, 2024k.

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Leon Bottou, Zhihao Jia, et al. Magicpig: Lsh sampling for efficient llm generation. *arXiv preprint arXiv:2410.16179*, 2024l.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788*, 2023.

Ta-Chung Chi, Ting-Han Fan, Peter J Ramadge, and Alexander Rudnicky. Kerple: Kernelized relative positional embedding for length extrapolation. *Advances in Neural Information Processing Systems*, 35:8386–8399, 2022.

Ta-Chung Chi, Ting-Han Fan, Li-Wei Chen, Alexander I Rudnicky, and Peter J Ramadge. Latent positional information is in the self-attention variance of transformer language models without positional embeddings. *arXiv preprint arXiv:2305.13571*, 2023.

Yew Ken Chia, Liying Cheng, Hou Pong Chan, Chaoqun Liu, Maojia Song, Sharifah Mahani Aljunied, Soujanya Poria, and Lidong Bing. M-longdoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. *arXiv preprint arXiv:2411.06176*, 2024.

Wonkyo Choe, Yangfeng Ji, and Felix Lin. Rwkv-edge: Deeply compressed rwkv for resource-constrained devices. *arXiv preprint arXiv:2412.10856*, 2024.

Eunseong Choi, Sunkyung Lee, Minjin Choi, June Park, and Jongwuk Lee. From reading to compressing: Exploring the multi-document reader for prompt compression. *arXiv preprint arXiv:2410.04139*, 2024.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *International Conference on Learning Representations*, 2020.

Shih-Han Chou, Shivam Chandhok, James J Little, and Leonid Sigal. Mm-r3: On (in-) consistency of multi-modal large language models (mllms). *arXiv preprint arXiv:2410.04778*, 2024.

Rohan Choudhury, Guanglei Zhu, Sihan Liu, Koichiro Niinuma, Kris M Kitani, and Laszlo Attila Jeni. Don't look twice: Faster video transformers with run-length tokenization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Tsz Ting Chung, Leyang Cui, Lemao Liu, Xinting Huang, Shuming Shi, and Dit-Yan Yeung. Selection-p: Self-supervised task-agnostic prompt compression for faithfulness and transferability. *arXiv preprint arXiv:2410.11786*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

ContextualAI. Introducing rag2, 2024. URL https://contextual.ai/introducing-rag2/.

LMDeploy Contributors. Lmdeploy: A toolkit for compressing, deploying, and serving llm. https://github.com/InternLM/lmdeploy, 2023.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, 2023.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Cade Daniel, Chen Shen, Eric Liang, and Richard Liaw. How continuous batching enables 23x throughput in llm inference while reducing p50 latency, 2024. URL https://www.anyscale.com/blog/continuous-batching-llm-inference.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024.

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.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4599–4610, 2021.

Jeffrey Dastin. Ai with reasoning power will be less predictable, ilya sutskever says, 2024. URL https://www.reuters.com/technology/artificial-intelligence/ai-with-reasoning-power-will-be-less-predictable-ilya-sutskever-says-2024-12-14/.

DeepSeek. Deepseek-r1-lite-preview is now live: unleashing supercharged reasoning power!, 2024. URL https://api-docs.deepseek.com/news/news1120.

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://github.com/deepseek-ai/DeepSeek-V3/blob/main/DeepSeek_V3.pdf. Accessed: 2024-12-26.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. A simple and effective l_2 norm-based strategy for kv cache compression. *arXiv preprint arXiv:2406.11430*, 2024.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. Longnet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*, 2023.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024.

Harry Dong, Tyler Johnson, Minsik Cho, and Emad Soroush. Towards low-bit communication for tensor parallel llm inference. *arXiv preprint arXiv:2411.07942*, 2024a. URL https://arxiv.org/abs/2411.07942.

Harry Dong, Xinyu Yang, Zhenyu Zhang, Zhangyang Wang, Yuejie Chi, and Beidi Chen. Get more with less: Synthesizing recurrence with kv cache compression for efficient llm inference. *arXiv preprint arXiv:2402.09398*, 2024b.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming model for generating optimized attention kernels. *arXiv preprint arXiv:2412.05496*, 2024c.

Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van Keirsbilck, Min-Hung Chen, Yoshi Suhara, et al. Hymba: A hybrid-head architecture for small language models. *arXiv preprint arXiv:2411.13676*, 2024d.

Zican Dong, Junyi Li, Xin Men, Wayne Xin Zhao, Bingbing Wang, Zhen Tian, Weipeng Chen, and Ji-Rong Wen. Exploring context window of large language models via decomposed positional vectors. *arXiv preprint arXiv:2405.18009*, 2024e.

Zican Dong, Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. Bamboo: A comprehensive benchmark for evaluating long text modeling capacities of large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 2086–2099, 2024f.

Jiangfei Duan, Shuo Zhang, Zerui Wang, Lijuan Jiang, Wenwen Qu, Qinghao Hu, Guoteng Wang, Qizhen Weng, Hang Yan, Xingcheng Zhang, et al. Efficient training of large language models on distributed infrastructures: A survey. *arXiv preprint arXiv:2407.20018*, 2024.

Haojie Duanmu, Zhihang Yuan, Xiuhong Li, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. Skvq: Sliding-window key and value cache quantization for large language models. *arXiv preprint arXiv:2405.06219*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pp. 75–92. Springer, 2025.

Jiarui Fang and Shangchun Zhao. Usp: A unified sequence parallelism approach for long context generative ai. *arXiv preprint arXiv:2405.07719*, 2024.

Junjie Fang, Likai Tang, Hongzhe Bi, Yujia Qin, Si Sun, Zhenyu Li, Haolun Li, Yongjian Li, Xin Cong, Yankai Lin, et al. Unimem: Towards a unified view of long-context large language models. *arXiv preprint arXiv:2402.03009*, 2024a.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. What is wrong with perplexity for long-context language modeling? *arXiv preprint arXiv:2410.23771*, 2024b.

Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao, Yining Li, Dahua Lin, and Kai Chen. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding. *arXiv preprint arXiv:2406.14515*, 2024c.

Weizhi Fei, Xueyan Niu, Pingyi Zhou, Lu Hou, Bo Bai, Lei Deng, and Wei Han. Extending context window of large language models via semantic compression. *arXiv preprint arXiv:2312.09571*, 2023.

Weizhi Fei, Xueyan Niu, Guoqing Xie, Yanhua Zhang, Bo Bai, Lei Deng, and Wei Han. Retrieval meets reasoning: Dynamic in-context editing for long-text understanding. *arXiv preprint arXiv:2406.12331*, 2024.

Leo Feng, Frederick Tung, Mohamed Osama Ahmed, Yoshua Bengio, and Hossein Hajimirsadegh. Were rnns all we needed? *arXiv preprint arXiv:2410.01201*, 2024.

FlashInfer Community. Flashinfer 0.2 - efficient and customizable kernels for llm inference serving, 2024. URL https://flashinfer.ai/2024/12/16/flashinfer-v02-release.html.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024a.

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Yao Fu. Challenges in deploying long-context transformers: A theoretical peak performance analysis. *arXiv preprint arXiv:2405.08944*, 2024.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context. In *Forty-first International Conference on Machine Learning*, 2024b.

Kazuki Fujii, Kohei Watanabe, and Rio Yokota. Accelerating large language model training with 4d parallelism and memory consumption estimator, 2024.

Bin Gao, Zhuomin He, Puru Sharma, Qingxuan Kang, Djordje Jevdjic, Junbo Deng, Xingkun Yang, Zhou Yu, and Pengfei Zuo. Attentionstore: Cost-effective attention reuse across multi-turn conversations in large language model serving. *arXiv preprint arXiv:2403.19708*, 2024a.

Chaochen Gao, Xing Wu, Qi Fu, and Songlin Hu. Quest: Query-centric data synthesis approach for long-context scaling of large language model. *arXiv preprint arXiv:2405.19846*, 2024b.

Mingze Gao, Jingyu Liu, Mingda Li, Jiangtao Xie, Qingbin Liu, Bo Zhao, Xi Chen, and Hui Xiong. Tc-llava: Rethinking the transfer from image to video understanding with temporal considerations. *arXiv preprint arXiv:2409.03206*, 2024c.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*, 2024d.

Shawn Gavin, Tuney Zheng, Jiaheng Liu, Quehry Que, Noah Wang, Jian Yang, Chenchen Zhang, Wenhao Huang, Wenhu Chen, and Ge Zhang. Longins: A challenging long-context instruction-based exam for llms. *arXiv preprint arXiv:2406.17588*, 2024.

Junqi Ge, Ziyi Chen, Jintao Lin, Jinguo Zhu, Xihui Liu, Jifeng Dai, and Xizhou Zhu. V2pe: Improving multimodal long-context capability of vision-language models with variable visual position encoding. *arXiv preprint arXiv:2412.09616*, 2024.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive kv cache compression for llms. *arXiv preprint arXiv:2310.01801*, 2023a.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023b.

Gerganov et al. llama.cpp, 2023. URL https://github.com/ggerganov/llama.cpp.

Amir Gholami, Zhewei Yao, Sehoon Kim, Coleman Hooper, Michael W Mahoney, and Kurt Keutzer. Ai and memory wall. *IEEE Micro*, 2024.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. *Proceedings of Machine Learning and Systems*, 6:325–338, 2024.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Shahriar Golchin and Mihai Surdeanu. Time travel in llms: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Omer Goldman, Alon Jacovi, Aviv Slobodkin, Aviya Maimon, Ido Dagan, and Reut Tsarfaty. Is it really long context if all you need is retrieval? towards genuinely difficult long context nlp. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16576–16586, 2024.

Daniel Goldstein, Fares Obeid, Eric Alcaide, Guangyu Song, and Eugene Cheah. Goldfinch: High performance rwkv/transformer hybrid with linear pre-fill and extreme kv-cache compression. *arXiv preprint arXiv:2407.12077*, 2024.

Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. Contextual position encoding: Learning to count what's important. *arXiv preprint arXiv:2405.18719*, 2024.

Google. Google pixel with gemini live, 2024a. URL https://store.google.com/ideas/gemini-ai-assistant.

Google. Gemini api long context, 2024b. URL https://ai.google.dev/gemini-api/docs/long-context.

Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

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

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, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021a.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021b.

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.

Diandian Gu, Peng Sun, Qinghao Hu, Ting Huang, Xun Chen, Yingtong Xiong, Guoteng Wang, Qiaoling Chen, Shangchun Zhao, Jiarui Fang, et al. Loongtrain: Efficient training of long-sequence llms with head-context parallelism. *arXiv preprint arXiv:2406.18485*, 2024a.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*, 2024b.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

Tom Gunter, Zirui Wang, Chong Wang, Ruoming Pang, Andy Narayanan, Aonan Zhang, Bowen Zhang, Chen Chen, Chung-Cheng Chiu, David Qiu, et al. Apple intelligence foundation language models. *arXiv preprint arXiv:2407.21075*, 2024.

Cong Guo, Feng Cheng, Zhixu Du, James Kiessling, Jonathan Ku, Shiyu Li, Ziru Li, Mingyuan Ma, Tergel Molom-Ochir, Benjamin Morris, et al. A survey: Collaborative hardware and software design in the era of large language models. *arXiv preprint arXiv:2410.07265*, 2024a.

Cong Guo, Rui Zhang, Jiale Xu, Jingwen Leng, Zihan Liu, Ziyu Huang, Minyi Guo, Hao Wu, Shouren Zhao, Junping Zhao, et al. Gmlake: Efficient and transparent gpu memory defragmentation for large-scale dnn training with virtual memory stitching. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pp. 450–466, 2024b.

Jialong Guo, Xinghao Chen, Yehui Tang, and Yunhe Wang. Slab: Efficient transformers with simplified linear attention and progressive re-parameterized batch normalization. *arXiv preprint arXiv:2405.11582*, 2024c.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. Attention score is not all you need for token importance indicator in kv cache reduction: Value also matters. *arXiv preprint arXiv:2406.12335*, 2024d.

Lavanya Gupta, Saket Sharma, and Yiyun Zhao. Systematic evaluation of long-context llms on financial concepts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1163–1175, 2024.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3991–4008, 2024.

Ramin Hasani, Mathias Lechner, Alexander Amini, Daniela Rus, and Radu Grosu. Liquid time-constant networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7657–7666, 2021.

Ramin Hasani, Mathias Lechner, Alexander Amini, Lucas Liebenwein, Aaron Ray, Max Tschaikowski, Gerald Teschl, and Daniela Rus. Closed-form continuous-time neural networks. *Nature Machine Intelligence*, 4(11):992–1003, 2022.

Demis Hassabis and Koray Kavukcuoglu. Introducing gemini 2.0: our new ai model for the agentic era, 2024. URL https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. Transformer language models without positional encodings still learn positional information. *arXiv preprint arXiv:2203.16634*, 2022.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13504–13514, 2024a.

Jiaao He and Jidong Zhai. Fastdecode: High-throughput gpu-efficient llm serving using heterogeneous pipelines. *arXiv preprint arXiv:2403.11421*, 2024.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipvl: Efficient large vision-language models with dynamic token sparsification and kv cache compression. *arXiv preprint arXiv:2410.08584*, 2024b.

Yefei He, Luoming Zhang, Weijia Wu, Jing Liu, Hong Zhou, and Bohan Zhuang. Zipcache: Accurate and efficient kv cache quantization with salient token identification. *arXiv preprint arXiv:2405.14256*, 2024c.

Zexue He, Leonid Karlinsky, Donghyun Kim, Julian McAuley, Dmitry Krotov, and Rogerio Feris. Camelot: Towards large language models with training-free consolidated associative memory. *arXiv preprint arXiv:2402.13449*, 2024d.

Ziwei He, Meng Yang, Minwei Feng, Jingcheng Yin, Xinbing Wang, Jingwen Leng, and Zhouhan Lin. Fourier transformer: Fast long range modeling by removing sequence redundancy with fft operator. *arXiv preprint arXiv:2305.15099*, 2023.

Lukas Hilgert, Danni Liu, and Jan Niehues. Evaluating and training long-context large language models for question answering on scientific papers. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pp. 220–236, 2024.

Connor Holmes, Masahiro Tanaka, Michael Wyatt, Ammar Ahmad Awan, Jeff Rasley, Samyam Rajbhandari, Reza Yazdani Aminabadi, Heyang Qin, Arash Bakhtiari, Lev Kurilenko, et al. Deepspeed-fastgen: High-throughput text generation for llms via mii and deepspeed-inference. *arXiv preprint arXiv:2401.08671*, 2024.

Ke Hong, Guohao Dai, Jiaming Xu, Qiuli Mao, Xiuhong Li, Jun Liu, Kangdi Chen, Yuhan Dong, and Yu Wang. Flashdecoding++: Faster large language model inference on gpus. *arXiv preprint arXiv:2311.01282*, 2023.

Xiangyu Hong, Che Jiang, Biqing Qi, Fandong Meng, Mo Yu, Bowen Zhou, and Jie Zhou. On the token distance modeling ability of higher rope attention dimension. *arXiv preprint arXiv:2410.08703*, 2024.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Monishwaran Maheswaran, June Paik, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. Squeezed attention: Accelerating long context length llm inference. *arXiv preprint arXiv:2411.09688*, 2024a.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *arXiv preprint arXiv:2401.18079*, 2024b.

Pedram Hosseini, Jessica M Sin, Bing Ren, Bryceton G Thomas, Elnaz Nouri, Ali Farahanchi, and Saeed Hassanpour. A benchmark for long-form medical question answering. In *Advancements In Medical Foundation Models: Explainability, Robustness, Security, and Beyond*, 2024a.

Pedram Hosseini, Jessica M Sin, Bing Ren, Bryceton G Thomas, Elnaz Nouri, Ali Farahanchi, and Saeed Hassanpour. A benchmark for long-form medical question answering. In *Advancements In Medical Foundation Models: Explainability, Robustness, Security, and Beyond*, 2024b.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024a.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, et al. Found in the middle: Calibrating positional attention bias improves long context utilization. *arXiv preprint arXiv:2406.16008*, 2024b.

Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. *arXiv preprint arXiv:2409.03420*, 2024a.

Cunchen Hu, Heyang Huang, Junhao Hu, Jiang Xu, Xusheng Chen, Tao Xie, Chenxi Wang, Sa Wang, Yungang Bao, Ninghui Sun, et al. Memserve: Context caching for disaggregated llm serving with elastic memory pool. *arXiv preprint arXiv:2406.17565*, 2024b.

Cunchen Hu, Heyang Huang, Liangliang Xu, Xusheng Chen, Jiang Xu, Shuang Chen, Hao Feng, Chenxi Wang, Sa Wang, Yungang Bao, et al. Inference without interference: Disaggregate llm inference for mixed downstream workloads. *arXiv preprint arXiv:2401.11181*, 2024c.

Jiawei Hu, Hong Jia, Mahbub Hassan, Lina Yao, Brano Kusy, and Wen Hu. Lightllm: A versatile large language model for predictive light sensing. *arXiv preprint arXiv:2411.15211*, 2024d.

Junhao Hu, Wenrui Huang, Haoyi Wang, Weidong Wang, Tiancheng Hu, Qin Zhang, Hao Feng, Xusheng Chen, Yizhou Shan, and Tao Xie. Epic: Efficient position-independent context caching for serving large language models. *arXiv preprint arXiv:2410.15332*, 2024e.

Qinghao Hu, Zhisheng Ye, Meng Zhang, Qiaoling Chen, Peng Sun, Yonggang Wen, and Tianwei Zhang. Hydro:{Surrogate-Based} hyperparameter tuning service in datacenters. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*, pp. 757–777, 2023.

Qinghao Hu, Zhisheng Ye, Zerui Wang, Guoteng Wang, Meng Zhang, Qiaoling Chen, Peng Sun, Dahua Lin, Xiaolin Wang, Yingwei Luo, et al. Characterization of large language model development in the datacenter. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pp. 709–729, 2024f.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024g.

Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, and Yansong Feng. Can perplexity reflect large language model's ability in long text understanding? In *The Second Tiny Papers Track at ICLR 2024*, 2024h.

Zhiyuan Hu, Yuliang Liu, Jinman Zhao, Suyuchen Wang, Wei Shen, Chao Yin, Bryan Hooi, et al. Cd-pos: Long context generalization in llms through continuous and discrete position synthesis. In *First Workshop on Long-Context Foundation Models@ ICML 2024*, 2024i.

Zhiyuan Hu, Yuliang Liu, Jinman Zhao, Suyuchen Wang, Yan Wang, Wei Shen, Qing Gu, Anh Tuan Luu, See-Kiong Ng, Zhiwei Jiang, et al. Longrecipe: Recipe for efficient long context generalization in large language models. *arXiv preprint arXiv:2409.00509*, 2024j.

Zhongzhe Hu, Junmin Xiao, Zheye Deng, Mingyi Li, Kewei Zhang, Xiaoyang Zhang, Ke Meng, Ninghui Sun, and Guangming Tan. Megtaichi: dynamic tensor-based memory management optimization for dnn training. *Proceedings of the 36th ACM International Conference on Supercomputing*, 2022.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1419–1436, 2021.

Mianqiu Huang, Xiaoran Liu, Shaojun Zhou, Mozhi Zhang, Chenkun Tan, Pengyu Wang, Qipeng Guo, Zhe Xu, Linyang Li, Zhikai Lei, et al. Longsafetybench: Long-context llms struggle with safety issues. *arXiv preprint arXiv:2411.06899*, 2024a.

Yunpeng Huang, Jingwei Xu, Junyu Lai, Zixu Jiang, Taolue Chen, Zenan Li, Yuan Yao, Xiaoxing Ma, Lijuan Yang, Hao Chen, et al. Advancing transformer architecture in long-context large language models: A comprehensive survey. *arXiv preprint arXiv:2311.12351*, 2023.

Yuxiang Huang, Binhang Yuan, Xu Han, Chaojun Xiao, and Zhiyuan Liu. Locret: Enhancing eviction in long-context llm inference with trained retaining heads. *arXiv preprint arXiv:2410.01805*, 2024b.

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. O1 replication journey–part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *arXiv preprint arXiv:2411.16489*, 2024c.

Hugging Face. Text generation inference, 2024. URL https://github.com/huggingface/text-generation-inference.

Mohamed Assem Ibrahim, Mahzabeen Islam, and Shaizeen Aga. Balanced data placement for gemv acceleration with processing-in-memory. *arXiv preprint arXiv:2403.20297*, 2024.

InternLM. Internlm2.5-7b, July 2024. URL https://huggingface.co/internlm/internlm2_5-7b.

InternLM. Internlm3-8b, January 2025. URL https://huggingface.co/internlm/internlm3-8b-instruct.

Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. In *European Conference on Computer Vision*, pp. 87–104. Springer, 2022.

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprinrt arXiv:2309.14509*, 2023.

Vijay Jaisankar, Sambaran Bandyopadhyay, Kalp Vyas, Varre Chaitanya, and Shwetha Somasundaram. Postdoc: Generating poster from a long multimodal document using deep submodular optimization. *arXiv preprint arXiv:2405.20213*, 2024.

Hongsun Jang, Jaeyong Song, Jaewon Jung, Jaeyoung Park, Youngsok Kim, and Jinho Lee. Smart-infinity: Fast large language model training using near-storage processing on a real system. In *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 345–360. IEEE, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023a.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*, 2023b.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, et al. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *arXiv preprint arXiv:2407.02490*, 2024a.

Xuanlin Jiang, Yang Zhou, Shiyi Cao, Ion Stoica, and Minlan Yu. Neo: Saving gpu memory crisis with cpu offloading for online llm inference. *arXiv preprint arXiv:2411.01142*, 2024b.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*, 2024c.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Chia-Yuan Chang, and Xia Hu. Growlength: Accelerating llms pretraining by progressively growing training length. *arXiv preprint arXiv:2310.00576*, 2023.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. Llm maybe longlm: Self-extend llm context window without tuning. *arXiv preprint arXiv:2401.01325*, 2024a.

Yibo Jin, Tao Wang, Huimin Lin, Mingyang Song, Peiyang Li, Yipeng Ma, Yicheng Shan, Zhengfan Yuan, Cailong Li, Yajing Sun, et al. P/d-serve: Serving disaggregated large language model at scale. *arXiv preprint arXiv:2408.08147*, 2024b.

Jordan Juravsky, Bradley Brown, Ryan Ehrlich, Daniel Y Fu, Christopher Ré, and Azalia Mirhoseini. Hydragen: High-throughput llm inference with shared prefixes. *arXiv preprint arXiv:2402.05099*, 2024.

Kumara Kahatapitiya, Kanchana Ranasinghe, Jongwoo Park, and Michael S Ryoo. Language repository for long video understanding. *arXiv preprint arXiv:2403.14622*, 2024.

Gabrielle Kaili-May Liu, Bowen Shi, Avi Caciularu, Idan Szpektor, and Arman Cohan. Mdcure: A scalable pipeline for multi-document instruction-following. *arXiv e-prints*, pp. arXiv–2410, 2024.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, et al. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

Greg Kamradt. Needle in a haystack - pressure testing llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023.

Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. Gear: An efficient kv cache compression recipefor near-lossless generative inference of llm. *arXiv preprint arXiv:2403.05527*, 2024.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. One thousand and one pairs: A "novel" challenge for long-context language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17048–17085, 2024.

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, 2024.

Wonkyun Kim, Changin Choi, Wonseok Lee, and Wonjong Rhee. An image grid can be worth a video: Zero-shot video question answering using a vlm. *arXiv preprint arXiv:2403.18406*, 2024.

Marisa Kirisame, Steven Lyubomirsky, Altan Haan, Jennifer Brennan, Mike He, Jared Roesch, Tianqi Chen, and Zachary Tatlock. Dynamic tensor rematerialization. *ArXiv*, abs/2006.09616, 2020.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. Longform: Effective instruction tuning with reverse instructions. *arXiv preprint arXiv:2304.08460*, 2023.

Seonmin Koo, Jinsung Kim, YoungJoon Jang, Chanjun Park, and Heui-Seok Lim. Where am i? large language models wandering between semantics and structures in long contexts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14144–14160, 2024.

Bruno Korbar, Yongqin Xian, Alessio Tonioni, Andrew Zisserman, and Federico Tombari. Text-conditioned resampler for long form video understanding. In *European Conference on Computer Vision*, pp. 271–288. Springer, 2025.

Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5:341–353, 2023.

Maurice Kraus, Felix Divo, Devendra Singh Dhami, and Kristian Kersting. xlstm-mixer: Multivariate time series forecasting by mixing via scalar memories. *arXiv preprint arXiv:2410.16928*, 2024.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *arXiv preprint arXiv:2406.10149*, 2024.

Wai-Chung Kwan, Xingshan Zeng, Yufei Wang, Yusen Sun, Liangyou Li, Lifeng Shang, Qun Liu, and Kam-Fai Wong. M4le: A multi-ability multi-range multi-task multi-domain long-context evaluation benchmark for large language models. *arXiv preprint arXiv:2310.19240*, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. Vidcompress: Memory-enhanced temporal compression for video understanding in large language models. *arXiv preprint arXiv:2410.11417*, 2024.

LangChain. Multi needle in a haystack. hhttps://blog.langchain.dev/multi-needle-in-a-haystack/, 2024.

Chris Lattner, Mehdi Amini, Uday Bondhugula, Albert Cohen, Andy Davis, Jacques Pienaar, River Riddle, Tatiana Shpeisman, Nicolas Vasilache, and Oleksandr Zinenko. Mlir: A compiler infrastructure for the end of moore's law. *arXiv preprint arXiv:2002.11054*, 2020.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien MR Arnold, Vincent Perot, Siddharth Dalmia, et al. Can long-context language models subsume retrieval, rag, sql, and more? *arXiv preprint arXiv:2406.13121*, 2024a.

Wonbeom Lee, Jungi Lee, Junghwan Seo, and Jaewoong Sim. {InfiniGen}: Efficient generative inference of large language models with dynamic {KV} cache management. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 155–172, 2024b.

Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, et al. Spider 2.0: Evaluating language models on real-world enterprise text-to-sql workflows. *arXiv preprint arXiv:2411.07763*, 2024a.

Fangyu Lei, Qian Liu, Yiming Huang, Shizhu He, Jun Zhao, and Kang Liu. S3eval: A synthetic, scalable, systematic evaluation suite for large language model. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1259–1286, 2024b.

Quinn Leng, Jacob Portes, Sam Havens, Matei Zaharia, and Michael Carbin. Long context rag performance of large language models. *arXiv preprint arXiv:2411.03538*, 2024.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models. *arXiv preprint arXiv:2402.14848*, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Baolin Li, Yankai Jiang, Vijay Gadepally, and Devesh Tiwari. Llm inference serving: Survey of recent advances and opportunities. *arXiv preprint arXiv:2407.12391*, 2024a.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024b.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph E. Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. How long can open-source llms truly promise on context length? Technical report, 2023a. URL https://lmsys.org/blog/2023-06-29-longchat/.

Dacheng Li, Rulin Shao, Anze Xie, Eric P Xing, Xuezhe Ma, Ion Stoica, Joseph E Gonzalez, and Hao Zhang. Distflashattn: Distributed memory-efficient attention for long-context llms training. In *First Conference on Language Modeling*, 2024c.

Jerry Li, Subhro Das, Aude Oliva, Dmitry Krotov, Leonid Karlinsky, and Rogerio Feris. Long context understanding using self-generated synthetic data. In *First Workshop on Long-Context Foundation Models@ ICML 2024*, NA.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. Loogle: Can long-context language models understand long contexts? *arXiv preprint arXiv:2311.04939*, 2023b.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36, 2024d.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023c.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, pp. 237–255. Springer, 2025a.

Lala Li and William Chan. Big bidirectional insertion representations for documents. *arXiv preprint arXiv:1910.13034*, 2019.

Lei Li, Yuanxin Liu, Linli Yao, Peiyuan Zhang, Chenxin An, Lean Wang, Xu Sun, Lingpeng Kong, and Qi Liu. Temporal reasoning transfer from text to video. *arXiv preprint arXiv:2410.06166*, 2024e.

Mo Li, Songyang Zhang, Yunxin Liu, and Kai Chen. Needlebench: Can llms do retrieval and reasoning in 1 million context window? *arXiv preprint arXiv:2407.11963*, 2024f.

Mukai Li, Lei Li, Shansan Gong, and Qi Liu. Giraffe: Design choices for extending the context length of visual language models. *arXiv preprint arXiv:2412.12735*, 2024g.

Shanda Li, Chong You, Guru Guruganesh, Joshua Ainslie, Santiago Ontanon, 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*, 2024h.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, et al. Pytorch distributed: Experiences on accelerating data parallel training. *arXiv preprint arXiv:2006.15704*, 2020.

Shenggui Li, Fuzhao Xue, Chaitanya Baranwal, Yongbin Li, and Yang You. Sequence parallelism: Long sequence training from system perspective. *arXiv preprint arXiv:2105.13120*, 2021a.

Siheng Li, Cheng Yang, Zesen Cheng, Lemao Liu, Mo Yu, Yujiu Yang, and Wai Lam. Large language models can self-improve in long-context reasoning. *arXiv preprint arXiv:2411.08147*, 2024i.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024j.

Wenhao Li, Mingbao Lin, Yunshan Zhong, Shuicheng Yan, and Rongrong Ji. Uio-llms: Unbiased incremental optimization for long-context llms. *arXiv preprint arXiv:2406.18173*, 2024k.

Xinze Li, Yixin Cao, Yubo Ma, and Aixin Sun. Long context vs. rag for llms: An evaluation and revisits. *arXiv preprint arXiv:2501.01880*, 2024l.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pp. 323–340. Springer, 2025b.

Yanyang Li, Shuo Liang, Michael R Lyu, and Liwei Wang. Making long-context language models better multi-hop reasoners. *arXiv preprint arXiv:2408.03246*, 2024m.

Youjie Li, Amar Phanishayee, Derek Murray, Jakub Tarnawski, and Nam Sung Kim. Harmony: Overcoming the hurdles of gpu memory capacity to train massive dnn models on commodity servers. *arXiv preprint arXiv:2202.01306*, 2022.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*, 2024n.

Zhan Li, Fanghui Liu, Volkan Cevher, and Grigorios Chrysos. Demonstrations in in-context learning for llms with large label space. In *First Workshop on Long-Context Foundation Models@ ICML 2024*, 2024o.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26763–26773, 2024p.

Zhenyu Li, Yike Zhang, Tengyu Pan, Yutao Sun, Zhichao Duan, Junjie Fang, Rong Han, Zixuan Wang, and Jianyong Wang. Focusllm: Scaling llm's context by parallel decoding. *arXiv preprint arXiv:2408.11745*, 2024q.

Zhuohan Li, Siyuan Zhuang, Shiyuan Guo, Danyang Zhuo, Hao Zhang, Dawn Xiaodong Song, and Ion Stoica. Terapipe: Token-level pipeline parallelism for training large-scale language models. In *International Conference on Machine Learning*, 2021b. URL https://api.semanticscholar.org/CorpusID:231934213.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 881–893, 2024r.

Hao Liang, Jiapeng Li, Tianyi Bai, Xijie Huang, Linzhuang Sun, Zhengren Wang, Conghui He, Bin Cui, Chong Chen, and Wentao Zhang. Keyvideollm: Towards large-scale video keyframe selection. *arXiv preprint arXiv:2407.03104*, 2024a.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, pp. 1–34, 2024b.

Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, Andrew Gu, Chien-Chin Huang, Iris Zhang, Wei Feng, Howard Huang, Junjie Wang, Sanket Purandare, Gokul Nadathur, and Stratos Idreos. Torchtitan: One-stop pytorch native solution for production ready llm pre-training. *arXiv preprint arXiv:2410.06511*, 2024c.

Yi Liang, You Wu, Honglei Zhuang, Li Chen, Jiaming Shen, Yiling Jia, Zhen Qin, Sumit Sanghai, Xuanhui Wang, Carl Yang, et al. Integrating planning into single-turn long-form text generation. *arXiv preprint arXiv:2410.06203*, 2024d.

Bingli Liao and Danilo Vasconcellos Vargas. Beyond kv caching: Shared attention for efficient llms. *arXiv preprint arXiv:2407.12866*, 2024.

Changyue Liao, Mo Sun, Zihan Yang, Kaiqi Chen, Binhang Yuan, Fei Wu, and Zeke Wang. Adding nvme ssds to enable and accelerate 100b model fine-tuning on a single gpu. *arXiv preprint arXiv:2403.06504*, 2024a.

Zihan Liao, Jun Wang, Hang Yu, Lingxiao Wei, Jianguo Li, and Wei Zhang. E2llm: Encoder elongated large language models for long-context understanding and reasoning. *arXiv preprint arXiv:2409.06679*, 2024b.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

LightLLM. Tokenattention, 2024. URL https://lightllm-en.readthedocs.io/en/latest/dev/token_attention.html.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations*, 2024a.

Bin Lin, Chen Zhang, Tao Peng, Hanyu Zhao, Wencong Xiao, Minmin Sun, Anmin Liu, Zhipeng Zhang, Lanbo Li, Xiafei Qiu, et al. Infinite-llm: Efficient llm service for long context with distattention and distributed kvcache. *arXiv preprint arXiv:2401.02669*, 2024b.

Bokai Lin, Zihao Zeng, Zipeng Xiao, Siqi Kou, Tianqi Hou, Xiaofeng Gao, Hao Zhang, and Zhijie Deng. Matryoshkakv: Adaptive kv compression via trainable orthogonal projection. *arXiv preprint arXiv:2410.14731*, 2024c.

Chaofan Lin, Zhenhua Han, Chengruidong Zhang, Yuqing Yang, Fan Yang, Chen Chen, and Lili Qiu. Parrot: Efficient serving of llm-based applications with semantic variable. *arXiv preprint arXiv:2405.19888*, 2024d.

Zhihui Lin, Maomao Li, Zhuobin Zheng, Yangyang Cheng, and Chun Yuan. Self-attention convlstm for spatiotemporal prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 11531–11538, 2020.

Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023.

Liquid. Liquid foundation models: Our first series of generative ai models, September 2024. URL https://www.liquid.ai/liquid-foundation-models.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024a.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. Minicache: Kv cache compression in depth dimension for large language models. *arXiv preprint arXiv:2405.14366*, 2024b.

Chenglong Liu, Haoran Wei, Jinyue Chen, Lingyu Kong, Zheng Ge, Zining Zhu, Liang Zhao, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Focus anywhere for fine-grained multi-page document understanding. *arXiv preprint arXiv:2405.14295*, 2024c.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, et al. Retrievalattention: Accelerating long-context llm inference via vector retrieval. *arXiv preprint arXiv:2409.10516*, 2024d.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023a.

Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with ringattention. *arXiv e-prints*, pp. arXiv–2402, 2024e.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024f.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024g. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024h.

Jiaheng Liu, Zhiqi Bai, Yuanxing Zhang, Chenchen Zhang, Yu Zhang, Ge Zhang, Jiakai Wang, Haoran Que, Yukang Chen, Wenbo Su, et al. E2-llm: Efficient and extreme length extension of large language models. *arXiv preprint arXiv:2401.06951*, 2024i.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*, 2024j.

Jiawei Liu, Jia Le Tian, Vijay Daita, Yuxiang Wei, Yifeng Ding, Yuhan Katherine Wang, Jun Yang, and Lingming Zhang. Repoqa: Evaluating long context code understanding. *arXiv preprint arXiv:2406.06025*, 2024k.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 11:157–173, 2024l.

Weijie Liu, Zecheng Tang, Juntao Li, Kehai Chen, and Min Zhang. Memlong: Memory-augmented retrieval for long text modeling. *arXiv preprint arXiv:2408.16967*, 2024m.

Xiang Liu, Peijie Dong, Xuming Hu, and Xiaowen Chu. Longgenbench: Long-context generation benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 865–883, 2024n.

Xiaoran Liu, Ruixiao Li, Qipeng Guo, Zhigeng Liu, Yuerong Song, Kai Lv, Hang Yan, Linlin Li, Qun Liu, and Xipeng Qiu. Reattention: Training-free infinite context with finite attention scope. *arXiv preprint arXiv:2407.15176*, 2024o.

Xiaoran Liu, Hang Yan, Chenxin An, Xipeng Qiu, and Dahua Lin. Scaling laws of rope-based extrapolation. In *The Twelfth International Conference on Learning Representations*, 2024p.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, et al. Deja vu: Contextual sparsity for efficient llms at inference time. In *International Conference on Machine Learning*, pp. 22137–22176. PMLR, 2023b.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for llm kv cache compression at test time. *Advances in Neural Information Processing Systems*, 36, 2024q.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*, 2024r.

Zuyan Liu, Benlin Liu, Jiahui Wang, Yuhao Dong, Guangyi Chen, Yongming Rao, Ranjay Krishna, and Jiwen Lu. Efficient inference of vision instruction-following models with elastic cache. In *European Conference on Computer Vision*, pp. 54–69. Springer, 2025.

Keer Lu, Xiaonan Nie, Zheng Liang, Da Pan, Shusen Zhang, Keshi Zhao, Weipeng Chen, Zenan Zhou, Guosheng Dong, Bin Cui, et al. Datasculpt: Crafting data landscapes for long-context llms through multi-objective partitioning. *arXiv preprint arXiv:2409.00997*, 2024a.

Xudong Lu, Yinghao Chen, Cheng Chen, Hui Tan, Boheng Chen, Yina Xie, Rui Hu, Guanxin Tan, Renshou Wu, Yan Hu, et al. Bluelm-v-3b: Algorithm and system co-design for multimodal large language models on mobile devices. *arXiv preprint arXiv:2411.10640*, 2024b.

Yi Lu, Xin Zhou, Wei He, Jun Zhao, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. Longheads: Multi-head attention is secretly a long context processor. *arXiv preprint arXiv:2402.10685*, 2024c.

Cheng Luo, Tianle Zhong, and Geoffrey Fox. Rtp: Rethinking tensor parallelism with memory deduplication. *arXiv preprint arXiv:2311.01635*, 2023.

Kun Luo, Zheng Liu, Shitao Xiao, and Kang Liu. Bge landmark embedding: A chunking-free embedding method for retrieval augmented long-context large language models. *arXiv preprint arXiv:2402.11573*, 2024.

Shi Luohe, Hongyi Zhang, Yao Yao, Zuchao Li, et al. Keep the cost down: A review on methods to optimize llm's kv-cache consumption. In *First Conference on Language Modeling*, 2024.

Kai Lv, Xiaoran Liu, Qipeng Guo, Hang Yan, Conghui He, Xipeng Qiu, and Dahua Lin. Longwanjuan: Towards systematic measurement for long text quality. *arXiv preprint arXiv:2402.13583*, 2024a.

Xingtai Lv, Ning Ding, Kaiyan Zhang, Ermo Hua, Ganqu Cui, and Bowen Zhou. Scalable efficient training of large language models with low-dimensional projected attention. *arXiv preprint arXiv:2411.02063*, 2024b.

Fan Ma, Xiaojie Jin, Heng Wang, Yuchen Xian, Jiashi Feng, and Yi Yang. Vista-llama: Reliable video narrator via equal distance to visual tokens. *arXiv preprint arXiv:2312.08870*, 2023.

Haiyue Ma, Jian Liu, and Ronny Krashinsky. Reducing the cost of dropout in flash-attention by hiding rng with gemm. *ArXiv*, abs/2410.07531, 2024a.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *arXiv preprint arXiv:2407.01523*, 2024b.

Seiji Maekawa, Hayate Iso, and Nikita Bhutani. Holistic reasoning with long-context lms: A benchmark for database operations on massive textual data. *arXiv preprint arXiv:2410.11996*, 2024.

Quentin Malartic, Nilabhra Roy Chowdhury, Ruxandra Cojocaru, Mugariya Farooq, Giulia Campesan, Yasser Abdelaziz Dahou Djilali, Sanath Narayan, Ankit Singh, Maksim Velikanov, Basma El Amel Boussaha, et al. Falcon2-11b technical report. *arXiv preprint arXiv:2407.14885*, 2024.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Kawshik Manikantan, Makarand Tapaswi, Vineet Gandhi, and Shubham Toshniwal. Identifyme: A challenging long-context mention resolution benchmark. *arXiv preprint arXiv:2411.07466*, 2024.

Reid McIlroy-Young, Katrina Brown, Conlan Olson, Linjun Zhang, and Cynthia Dwork. Order-independence without fine tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Xin Men, Mingyu Xu, Bingning Wang, Qingyu Zhang, Hongyu Lin, Xianpei Han, and Weipeng Chen. Base of rope bounds context length. *arXiv preprint arXiv:2405.14591*, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2022.

AI Meta. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI.*, 2024a.

AI Meta. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI.*, 2024b.

MiniMax. abab7-preview, 2024. URL https://www.minimaxi.com/news/abab7-preview%E5%8F%91%E5%B8%83.

MiniMax, Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, Enwei Jiao, Gengxin Li, Guojun Zhang, Haohai Sun, Houze Dong, Jiadai Zhu, Jiaqi Zhuang, Jiayuan Song, Jin Zhu, Jingtao Han, Jingyang Li, Junbin Xie, Junhao Xu, Junjie Yan, Kaishun Zhang, Kecheng Xiao, Kexi Kang, Le Han, Leyang Wang, Lianfei Yu, Liheng Feng, Lin Zheng, Linbo Chai, Long Xing, Meizhi Ju, Mingyuan Chi, Mozhi Zhang, Peikai Huang, Pengcheng Niu, Pengfei Li, Pengyu Zhao, Qi Yang, Qidi Xu, Qiexiang Wang, Qin Wang, Qiuhui Li, Ruitao Leng, Shengmin Shi, Shuqi Yu, Sichen Li, Songquan Zhu, Tao Huang, Tianrun Liang, Weigao Sun, Weixuan Sun, Weiyu Cheng, Wenkai Li, Xiangjun Song, Xiao Su, Xiaodong Han, Xinjie Zhang, Xinzhu Hou, Xu Min, Xun Zou, Xuyang Shen, Yan Gong, Yingjie Zhu, Yipeng Zhou, Yiran Zhong, Yongyi Hu, Yuanxiang Fan, Yue Yu, Yufeng Yang, Yuhao Li, Yunan Huang, Yunji Li, Yunpeng Huang, Yunzhi Xu, Yuxin Mao, Zehan Li, Zekang Li, Zewei Tao, Zewen Ying, Zhaoyang Cong, Zhen Qin, Zhenhua Fan, Zhihang Yu, Zhuo Jiang, and Zijia Wu. Minimax-01: Scaling foundation models with lightning attention, 2025. URL https://arxiv.org/abs/2501.08313.

MLC team. MLC-LLM, 2023-2024. URL https://github.com/mlc-ai/mlc-llm.

Amirkeivan Mohtashami and Martin Jaggi. Random-access infinite context length for transformers. *Advances in Neural Information Processing Systems*, 36:54567–54585, 2023.

MoonshotAI. Moonshotai kimi, 2023. URL https://mp.weixin.qq.com/s/V4FUdkqUm2erGNliQt9WuA.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 12284–12314, 2023.

Ivona Najdenkoska, Mohammad Mahdi Derakhshani, Yuki M Asano, Nanne van Noord, Marcel Worring, and Cees GM Snoek. Tulip: Token-length upgraded clip. *arXiv preprint arXiv:2410.10034*, 2024.

Sharan Narang, Gregory Diamos, Erich Elsen, Paulius Micikevicius, Jonah Alben, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *Int. Conf. on Learning Representation*, 2017.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Anand Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei A. Zaharia. Efficient large-scale language model training on gpu clusters using megatron-lm. *SC21: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–14, 2021.

Piotr Nawrot, Adrian Łańcucki, Marcin Chochowski, David Tarjan, and Edoardo M Ponti. Dynamic memory compression: Retrofitting llms for accelerated inference. *arXiv preprint arXiv:2403.09636*, 2024.

Eric Nguyen, Karan Goel, Albert Gu, Gordon Downs, Preey Shah, Tri Dao, Stephen Baccus, and Christopher Ré. S4nd: Modeling images and videos as multidimensional signals with state spaces. *Advances in neural information processing systems*, 35:2846–2861, 2022.

Xuanfan Ni, Hengyi Cai, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, and Piji Li. Xl2bench: A benchmark for extremely long context understanding with long-range dependencies. *arXiv preprint arXiv:2404.05446*, 2024.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2241–2252, 2017.

NVIDIA. Transformer engine, 2024a. URL https://github.com/NVIDIA/TransformerEngine.

NVIDIA, 2024b. URL https://github.com/NVIDIA/TensorRT-LLM.

OpenAI. Chatgpt: Advanced language model by openai, 2022. URL https://openai.com/index/chatgpt/.

OpenAI. Gpt-4 technical report. Technical report, 2023a.

OpenAI. tiktoken: A fast bpe tokeniser for use with openai's models. https://github.com/openai/tiktoken, 2023b.

OpenAI. O1: Openai's first model, 2024. URL https://openai.com/o1/. Accessed: 2024-12-25.

OpenLMLab, 2024. URL https://github.com/OpenLMLab/ParallelTokenizer.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are multi-state rnns. *arXiv preprint arXiv:2401.06104*, 2024.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

Arka Pal, Deep Karkhanis, Manley Roberts, Samuel Dooley, Arvind Sundararajan, and Siddartha Naidu. Giraffe: Adventures in expanding context lengths in llms. *arXiv preprint arXiv:2308.10882*, 2023.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8298–8319, 2023.

Jianhui Pang, Fanghua Ye, Derek Fai Wong, Xin He, Wanshun Chen, and Longyue Wang. Anchor-based large language models. *arXiv preprint arXiv:2402.07616*, 2024.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. Quality: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5336–5358, 2022.

Jongwoo Park, Kanchana Ranasinghe, Kumara Kahatapitiya, Wonjeong Ryoo, Donghyun Kim, and Michael S Ryoo. Too many frames, not all useful: Efficient strategies for long-form video qa. *arXiv preprint arXiv:2406.09396*, 2024.

Dylan Patel and Daniel Nishball. Nvidia blackwell perf tco analysis – b100 vs b200 vs gb200 nvl72, 2024. URL https://semianalysis.com/2024/04/10/nvidia-blackwell-perf-tco-analysis/.

Dylan Patel, Myron Xie, and Gerald Wong. Ai capacity constraints – cowos and hbm supply chain, 2023. URL https://semianalysis.com/2023/07/05/ai-capacity-constraints-cowos-and/.

Dylan Patel, Daniel Nishball, and Reyk Knuhtsen. Amazon's ai self sufficiency — trainium2 architecture & networking, 2024a. URL https://semianalysis.com/2024/12/03/amazons-ai-self-sufficiency-trainium2-architecture-networking/.

Dylan Patel, Daniel Nishball, and AJ Kourabi. Scaling laws – o1 pro architecture, reasoning training infrastructure, orion and claude 3.5 opus "failures", 2024b. URL https://semianalysis.com/2024/12/11/scaling-laws-o1-pro-architecture-reasoning-training-infrastructure-orion-and-claude-3-5-opus-failures.

Pratyush Patel, Esha Choukse, Chaojie Zhang, Aashaka Shah, Íñigo Goiri, Saeed Maleki, and Ricardo Bianchini. Splitwise: Efficient generative llm inference using phase splitting. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pp. 118–132. IEEE, 2024c.

Saurav Pawar, SM Tonmoy, SM Zaman, Vinija Jain, Aman Chadha, and Amitava Das. The what, why, and how of context length extension techniques in large language models–a detailed survey. *arXiv preprint arXiv:2401.07872*, 2024.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023a.

Bo Peng, Daniel Goldstein, Quentin Anthony, Alon Albalak, Eric Alcaide, Stella Biderman, Eugene Cheah, Xingjian Du, Teddy Ferdinan, Haowen Hou, et al. Eagle and finch: Rwkv with matrix-valued states and dynamic recurrence. *arXiv preprint arXiv:2404.05892*, 2024a.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*, 2024b.

Hao Peng, Jungo Kasai, Nikolaos Pappas, Dani Yogatama, Zhaofeng Wu, Lingpeng Kong, Roy Schwartz, and Noah A Smith. Abc: Attention with bounded-memory control. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Houwen Peng, Kan Wu, Yixuan Wei, Guoshuai Zhao, Yuxiang Yang, Ze Liu, Yifan Xiong, Ziyue Yang, Bolin Ni, Jingcheng Hu, et al. Fp8-lm: Training fp8 large language models. *arXiv preprint arXiv:2310.18313*, 2023b.

Cory Perry and Nikolay Sakharnykh, 2024. URL https://developer.nvidia.com/blog/introducing-low-level-gpu-virtual-memory-management/.

Chau Minh Pham, Simeng Sun, and Mohit Iyyer. Suri: Multi-constraint instruction following for long-form text generation. *arXiv preprint arXiv:2406.19371*, 2024.

Alessandro Pierro and Steven Abreu. Mamba-ptq: Outlier channels in recurrent large language models. *arXiv preprint arXiv:2407.12397*, 2024.

Ramya Prabhu, Ajay Nayak, Jayashree Mohan, Ramachandran Ramjee, and Ashish Panwar. vattention: Dynamic memory management for serving llms without pagedattention. *arXiv preprint arXiv:2405.04437*, 2024.

Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022.

Bharadwaj Pudipeddi, Maral Mesmakhosroshahi, Jinwen Xi, and Sujeeth Bharadwaj. Training large neural networks with constant memory using a new execution algorithm. *arXiv preprint arXiv:2002.05645*, 2020.

PyTorch, 2024. URL https://pytorch.org/docs/stable/notes/cuda.html.

PyTorch. Maximizing training throughput using pytorch fsdp, 2024. URL https://pytorch.org/blog/maximizing-training/#selective-activation-checkpointing. Accessed: 2024-3-13.

Biqing Qi, Junqi Gao, Kaiyan Zhang, Dong Li, Jianxing Liu, Ligang Wu, and Bowen Zhou. Smr: State memory replay for long sequence modeling. *arXiv preprint arXiv:2405.17534*, 2024a.

Penghui Qi, Xinyi Wan, Guangxing Huang, and Min Lin. Zero bubble (almost) pipeline parallelism. In *International Conference on Learning Representations*, 2024b. URL https://api.semanticscholar.org/CorpusID:270799166.

Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. In *Forty-first International Conference on Machine Learning*, 2024a.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *arXiv preprint arXiv:2405.16009*, 2024b.

Aurick Qiao, Zhewei Yao, Samyam Rajbhandari, and Yuxiong He. Swiftkv: Fast prefill-optimized inference with knowledge-preserving model transformation. *arXiv preprint arXiv:2410.03960*, 2024.

Ruoyu Qin, Zheming Li, Weiran He, Mingxing Zhang, Yongwei Wu, Weimin Zheng, and Xinran Xu. Mooncake: A kvcache-centric disaggregated architecture for llm serving. *arXiv preprint arXiv:2407.00079*, 2024a.

Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, et al. O1 replication journey: A strategic progress report–part 1. *arXiv preprint arXiv:2410.18982*, 2024b.

Zhen Qin, Weigao Sun, Dong Li, Xuyang Shen, Weixuan Sun, and Yiran Zhong. Lightning attention-2: A free lunch for handling unlimited sequence lengths in large language models. *arXiv preprint arXiv:2401.04658*, 2024c.

Zhen Qin, Weigao Sun, Dong Li, Xuyang Shen, Weixuan Sun, and Yiran Zhong. Various lengths, constant speed: Efficient language modeling with lightning attention. *arXiv preprint arXiv:2405.17381*, 2024d.

Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. Hgrn2: Gated linear rnns with state expansion. *arXiv preprint arXiv:2404.07904*, 2024e.

Zhen Qin, Songlin Yang, and Yiran Zhong. Hierarchically gated recurrent neural network for sequence modeling. *Advances in Neural Information Processing Systems*, 36, 2024f.

Zexuan Qiu, Jingjing Li, Shijue Huang, Xiaoqi Jiao, Wanjun Zhong, and Irwin King. Clongeval: A chinese benchmark for evaluating long-context large language models. *arXiv preprint arXiv:2403.03514*, 2024.

Qualcomm. Qualcomm works with meta to enable on-device ai applications using llama 2, 2023. URL https://www.qualcomm.com/news/releases/2023/07/qualcomm-works-with-meta-to-enable-on-device-ai-applications-usi.

Qualcomm. Qualcomm and mistral ai partner to bring new generative ai models to edge devices, 2024. URL https://www.qualcomm.com/news/releases/2024/10/qualcomm-and-mistral-ai-partner-to-bring-new-generative-ai-model.

Shanghaoran Quan, Tianyi Tang, Bowen Yu, An Yang, Dayiheng Liu, Bofei Gao, Jianhong Tu, Yichang Zhang, Jingren Zhou, and Junyang Lin. Language models can self-lengthen to generate long texts. *arXiv preprint arXiv:2410.23933*, 2024.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, et al. Hellobench: Evaluating long text generation capabilities of large language models. *arXiv preprint arXiv:2409.16191*, 2024.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, Chloe Hillier, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, and Yuxiong He. Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. In *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pp. 1–14, 2021.

Venkat Raman. Essential math & concepts for llm inference, 2024. URL https://venkat.eu/essential-math-concepts-for-llm-inference#heading-insights-from-model-latency-amp-understanding-hardware-utilization-on-modern-gpus.

Kanchana Ranasinghe, Xiang Li, Kumara Kahatapitiya, and Michael S Ryoo. Understanding long videos in one multimodal language model pass. *arXiv preprint arXiv:2403.16998*, 2024.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. Parallel context windows for large language models. *arXiv preprint arXiv:2212.10947*, 2022.

Varshini Reddy, Rik Koncel-Kedziorski, Viet Dac Lai, Michael Krumdick, Charles Lovering, and Chris Tanner. Docfinqa: A long-context financial reasoning dataset. *arXiv preprint arXiv:2401.06915*, 2024.

Isaac Rehg. Kv-compress: Paged kv-cache compression with variable compression rates per attention head. *arXiv preprint arXiv:2410.00161*, 2024.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. {Zero-offload}: Democratizing {billion-scale} model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pp. 551–564, 2021.

Shuhuai Ren, Sishuo Chen, Shicheng Li, Xu Sun, and Lu Hou. Testa: Temporal-spatial token aggregation for long-form video-language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 932–947, 2023a.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14313–14323, 2024a.

Siyu Ren and Kenny Q Zhu. On the efficacy of eviction policy for key-value constrained generative language model inference. *arXiv preprint arXiv:2402.06262*, 2024.

Siyu Ren, Qi Jia, and Kenny Q Zhu. Context compression for auto-regressive transformers with sentinel tokens. *arXiv preprint arXiv:2310.08152*, 2023b.

Weiming Ren, Huan Yang, Jie Min, Cong Wei, and Wenhu Chen. Vista: Enhancing long-duration and high-resolution video understanding by video spatiotemporal augmentation. *arXiv preprint arXiv:2412.00927*, 2024b.

Luka Ribar, Ivan Chelombiev, Luke Hudlass-Galley, Charlie Blake, Carlo Luschi, and Douglas Orr. Sparq attention: Bandwidth-efficient llm inference.(2023). *arXiv preprint cs.LG/2312.04985*, 2023.

Jonathan Roberts, Kai Han, and Samuel Albanie. Needle threading: Can llms follow threads through near-million-scale haystacks? *arXiv preprint arXiv:2411.05000*, 2024.

Kashob Kumar Roy, Pritom Saha Akash, Kevin Chen-Chuan Chang, and Lucian Popa. Contregen: Context-driven tree-structured retrieval for open-domain long-form text generation. *arXiv preprint arXiv:2410.15511*, 2024.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Anian Ruoss, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Róbert Csordás, Mehdi Bennani, Shane Legg, and Joel Veness. Randomized positional encodings boost length generalization of transformers. *arXiv preprint arXiv:2305.16843*, 2023.

Ali Safaya and Deniz Yuret. Neurocache: Efficient vector retrieval for long-range language modeling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 870–883, 2024.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for kv cache compression. *arXiv preprint arXiv:2408.05646*, 2024.

Jürgen Schmidhuber, Sepp Hochreiter, et al. Long short-term memory. *Neural Comput*, 9(8): 1735–1780, 1997.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *arXiv preprint arXiv:2407.08608*, 2024.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, et al. Scrolls: Standardized comparison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 12007–12021, 2022.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. Zeroscrolls: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 7977–7989, 2023.

Yuzhang Shang, Bingxin Xu, Weitai Kang, Mu Cai, Yuheng Li, Zehao Wen, Zhen Dong, Kurt Keutzer, et al. Interpolating video-llms: Toward longer-sequence lmms in a training-free manner. *arXiv preprint arXiv:2409.12963*, 2024.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Ré, Ion Stoica, and Ce Zhang. Flexgen: High-throughput generative inference of large language models with a single gpu. In *International Conference on Machine Learning*, pp. 31094–31116. PMLR, 2023.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A Smith, Luke Zettlemoyer, Wen-tau Yih, and Mike Lewis. In-context pretraining: Language modeling beyond document boundaries. In *The Twelfth International Conference on Learning Representations*, 2024.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *ArXiv*, abs/1909.08053, 2019. URL https://api.semanticscholar.org/CorpusID:202660670.

Huiyao Shu, Ang Wang, Ziji Shi, Hanyu Zhao, Yong Li, and Lu Lu. Roam: memory-efficient large dnn training via optimized operator ordering and memory layout. *arXiv preprint arXiv:2310.19295*, 2023.

Shuzheng Si, Haozhe Zhao, Gang Chen, Yunshui Li, Kangyang Luo, Chuancheng Lv, Kaikai An, Fanchao Qi, Baobao Chang, and Maosong Sun. Selecting influential samples for long context alignment via homologous models' guidance and contextual awareness measurement. *arXiv preprint arXiv:2410.15633*, 2024.

Siddharth Singh, Prajwal Singhania, Aditya K Ranjan, Zack Sating, and Abhinav Bhatele. A 4d hybrid algorithm to scale parallel training to thousands of gpus. *arXiv preprint arXiv:2305.13525*, 2024.

Prajwal Singhania, Siddharth Singh, Shwai He, Soheil Feizi, and Abhinav Bhatele. Loki: Low-rank keys for efficient sparse attention. *arXiv preprint arXiv:2406.02542*, 2024.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Yun Joon Soh, Hanxian Huang, Yuandong Tian, and Jishen Zhao. You only use reactive attention slice for long context retrieval. *arXiv preprint arXiv:2409.13695*, 2024.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18221–18232, 2024a.

Kaiqiang Song, Xiaoyang Wang, Sangwoo Cho, Xiaoman Pan, and Dong Yu. Zebra: Extending context window with layerwise grouped local-global attention. *arXiv preprint arXiv:2312.08618*, 2023.

Mingyang Song, Mao Zheng, and Xuan Luo. Counting-stars: A multi-evidence, position-aware, and scalable benchmark for evaluating long-context large language models. *arXiv preprint arXiv:2403.11802*, 2024b.

Yixin Song, Zeyu Mi, Haotong Xie, and Haibo Chen. Powerinfer: Fast large language model serving with a consumer-grade gpu. In *Proceedings of the ACM SIGOPS 30th Symposium on Operating Systems Principles*, pp. 590–606, 2024c.

Benjamin F. Spector, Simran Arora, Aaryan Singhal, Daniel Y. Fu, and Christopher R'e. Thunderkittens: Simple, fast, and adorable ai kernels. *ArXiv*, abs/2410.20399, 2024.

Vikranth Srivatsa, Zijian He, Reyna Abhyankar, Dongming Li, and Yiying Zhang. Preble: Efficient distributed prompt scheduling for llm serving. *arXiv preprint arXiv:2407.00023*, 2024.

Konrad Staniszewski, Szymon Tworkowski, Sebastian Jaszczur, Yu Zhao, Henryk Michalewski, Łukasz Kuciński, and Piotr Miłoś. Structured packing in llm training improves long context utilization. *arXiv preprint arXiv:2312.17296*, 2023.

Jianlin Su. Nbce: Naive bayes-based context extension, May 2023a. URL https://kexue.fm/archives/9617.

Jianlin Su. Rerope: Rectified rotary position embeddings, July 2023b. URL https://github.com/bojone/rerope.

Jianlin Su. Transformer upgrade path: 17. insights into multimodal positional encoding, March 2024. URL https://spaces.ac.cn/archives/10040.

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

Ao Sun, Weilin Zhao, Xu Han, Cheng Yang, Xinrong Zhang, Zhiyuan Liu, Chuan Shi, and Maosong Sun. Seq1f1b: Efficient sequence-level pipeline parallelism for large language model training. *arXiv preprint arXiv:2406.03488*, 2024a.

Biao Sun, Ziming Huang, Hanyu Zhao, Wencong Xiao, Xinyi Zhang, Yong Li, and Wei Lin. Llumnix: Dynamic scheduling for large language model serving. *arXiv preprint arXiv:2406.03243*, 2024b.

Tianxiang Sun, Xiaotian Zhang, Zhengfu He, Peng Li, Qinyuan Cheng, Xiangyang Liu, Hang Yan, Yunfan Shao, Qiong Tang, Shiduo Zhang, Xingjian Zhao, Ke Chen, Yining Zheng, Zhejian Zhou, Ruixiao Li, Jun Zhan, Yunhua Zhou, Linyang Li, Xiaogui Yang, Lingling Wu, Zhangyue Yin, Xuanjing Huang, Yu-Gang Jiang, and Xipeng Qiu. Moss: An open conversational large language model. *Machine Intelligence Research*, 2024c. ISSN 2731-5398. doi: 10.1007/s11633-024-1502-8. URL https://github.com/OpenMOSS/MOSS.

Weigao Sun, Zhen Qin, Dong Li, Xuyang Shen, Yu Qiao, and Yiran Zhong. Linear attention sequence parallelism, 2024d.

Weigao Sun, Zhen Qin, Weixuan Sun, Shidi Li, Dong Li, Xuyang Shen, Yu Qiao, and Yiran Zhong. Co2: Efficient distributed training with full communication-computation overlap. *arXiv preprint arXiv:2401.16265*, 2024e.

Xiao Sun, Jungwook Choi, Chia-Yu Chen, Naigang Wang, Swagath Venkataramani, Vijayalakshmi Viji Srinivasan, Xiaodong Cui, Wei Zhang, and Kailash Gopalakrishnan. Hybrid 8-bit floating point (hfp8) training and inference for deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Xiaoyang Sun, Wei Wang, Shenghao Qiu, Renyu Yang, Songfang Huang, Jie Xu, and Zheng Wang. Stronghold: fast and affordable billion-scale deep learning model training. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–17. IEEE, 2022a.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024f.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. *arXiv preprint arXiv:2212.10554*, 2022b.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Yutao Sun, Li Dong, Yi Zhu, Shaohan Huang, Wenhui Wang, Shuming Ma, Quanlu Zhang, Jianyong Wang, and Furu Wei. You only cache once: Decoder-decoder architectures for language models. *arXiv preprint arXiv:2405.05254*, 2024g.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36, 2024h.

Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

Haochen Tan, Zhijiang Guo, Zhan Shi, Lu Xu, Zhili Liu, Yunlong Feng, Xiaoguang Li, Yasheng Wang, Lifeng Shang, Qun Liu, et al. Proxyqa: An alternative framework for evaluating long-form text generation with large language models. *arXiv preprint arXiv:2401.15042*, 2024a.

Sijun Tan, Xiuyu Li, Shishir Patil, Ziyang Wu, Tianjun Zhang, Kurt Keutzer, Joseph E Gonzalez, and Raluca Ada Popa. Lloco: Learning long contexts offline. *arXiv preprint arXiv:2404.07979*, 2024b.

Hanlin Tang, Yang Lin, Jing Lin, Qingsen Han, Shikuan Hong, Yiwu Yao, and Gongyi Wang. Razorattention: Efficient kv cache compression through retrieval heads. *arXiv preprint arXiv:2407.15891*, 2024a.

Jiwei Tang, Jin Xu, Tingwei Lu, Zhicheng Zhang, Yiming Zhao, Lin Hai, and Hai-Tao Zheng. Perception compressor: A training-free prompt compression method in long context scenarios. *arXiv preprint arXiv:2409.19272*, 2024b.

Zecheng Tang, Zechen Sun, Juntao Li, Qiaoming Zhu, and Min Zhang. Logo–long context alignment via efficient preference optimization. *arXiv preprint arXiv:2410.18533*, 2024c.

Zecheng Tang, Keyan Zhou, Juntao Li, Baibei Ji, Jianye Hou, and Min Zhang. L-citeeval: Do long-context models truly leverage context for responding? *arXiv preprint arXiv:2410.02115*, 2024d.

Qian Tao, Wenyuan Yu, and Jingren Zhou. Asymkv: Enabling 1-bit quantization of kv cache with layer-wise asymmetric quantization configurations. *arXiv preprint arXiv:2410.13212*, 2024.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. In *International Conference on Learning Representations*, 2021.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024a.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024b.

Jamba Team, Barak Lenz, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, et al. Jamba-1.5: Hybrid transformer-mamba models at scale. *arXiv preprint arXiv:2408.12570*, 2024c.

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024a. URL https://qwenlm.github.io/blog/qwq-32b-preview/.

Step Team. Step-1v, 2024b. URL https://www.stepfun.com/#step1v.

TII Team. The falcon 3 family of open models, December 2024c.

Baichuan Intelligent Technology. Baichuan-7b: An open-source large-scale pre-trained model, 2023. URL https://huggingface.co/baichuan-inc/Baichuan-7B. Accessed: 2025-01-10.

Thibaut Thonet, Jos Rozen, and Laurent Besacier. Elitr-bench: A meeting assistant benchmark for long-context language models. *arXiv preprint arXiv:2403.20262*, 2024.

Junfeng Tian, Da Zheng, Yang Cheng, Rui Wang, Colin Zhang, and Debing Zhang. Untie the knots: An efficient data augmentation strategy for long-context pre-training in language models. *arXiv preprint arXiv:2409.04774*, 2024.

Philippe Tillet, Hsiang-Tsung Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pp. 10–19, 2019.

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*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. Vl-cache: Sparsity and modality-aware kv cache compression for vision-language model inference acceleration. *arXiv preprint arXiv:2410.23317*, 2024.

Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. Focused transformer: Contrastive training for context scaling. *Advances in Neural Information Processing Systems*, 36, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Mathurin Videau, Badr Youbi Idrissi, Daniel Haziza, Luca Wehrstedt, Jade Copet, Olivier Teytaud, and David Lopez-Paz. Meta Lingua: A minimal PyTorch LLM training library, 2024. URL https://github.com/facebookresearch/lingua.

Kiran Vodrahalli, Santiago Ontanon, Nilesh Tripuraneni, Kelvin Xu, Sanil Jain, Rakesh Shivanna, Jeffrey Hui, Nishanth Dikkala, Mehran Kazemi, Bahare Fatemi, et al. Michelangelo: Long context evaluations beyond haystacks via latent structure queries. *arXiv preprint arXiv:2409.12640*, 2024.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4065–4078, 2024.

Chonghua Wang, Haodong Duan, Songyang Zhang, Dahua Lin, and Kai Chen. Ada-leval: Evaluating long-context llms with length-adaptable benchmarks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3712–3724, 2024a.

Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Qian Wang, and Yue Zhang. Novelqa: A benchmark for long-range novel question answering. *arXiv preprint arXiv:2403.12766*, 2024b.

Fan Wang, Chuan Lin, Yang Cao, and Yu Kang. Benchmarking general purpose in-context learning. *arXiv preprint arXiv:2405.17234*, 2024c.

Guanhua Wang, Heyang Qin, Sam Ade Jacobs, Connor Holmes, Samyam Rajbhandari, Olatunji Ruwase, Feng Yan, Lei Yang, and Yuxiong He. Zero++: Extremely efficient collective communication for giant model training. *arXiv preprint arXiv:2306.10209*, 2023a.

Haiquan Wang, Chaoyi Ruan, Jia He, Jiaqi Ruan, Chengjie Tang, Xiaosong Ma, and Cheng Li. Hiding communication cost in distributed llm training via micro-batch co-execution. *arXiv preprint arXiv:2411.15871*, 2024d.

Haonan Wang, Qian Liu, Chao Du, Tongyao Zhu, Cunxiao Du, Kenji Kawaguchi, and Tianyu Pang. When precision meets position: Bfloat16 breaks down rope in long-context training. *arXiv preprint arXiv:2411.13476*, 2024e.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*, 2023b.

Jiapeng Wang, Chengyu Wang, Kunzhe Huang, Jun Huang, and Lianwen Jin. Videoclip-xl: Advancing long description understanding for video clip models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16061–16075, 2024f.

Jie Wang, Tao Ji, Yuanbin Wu, Hang Yan, Tao Gui, Qi Zhang, Xuanjing Huang, and Xiaoling Wang. Length generalization of causal transformers without position encoding. *arXiv preprint arXiv:2404.12224*, 2024g.

Jue Wang, Wentao Zhu, Pichao Wang, Xiang Yu, Linda Liu, Mohamed Omar, and Raffay Hamid. Selective structured state-spaces for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6387–6397, 2023c.

Junxiong Wang, Daniele Paliotta, Avner May, Alexander M Rush, and Tri Dao. The mamba in the llama: Distilling and accelerating hybrid models. *arXiv preprint arXiv:2408.15237*, 2024h.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, et al. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5627–5646, 2024i.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2023d.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024j.

Shengnan Wang, Youhui Bai, Lin Zhang, Pingyi Zhou, Shixiong Zhao, Gong Zhang, Sen Wang, Renhai Chen, Hua Xu, and Hongwei Sun. Xl3m: A training-free framework for llm length extension based on segment-wise inference. *arXiv preprint arXiv:2405.17755*, 2024k.

Siping Wang. Fastgemv, 2023. URL https://github.com/wangsiping97/FastGEMV.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pp. 58–76. Springer, 2025.

Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via a hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2024l.

Y Wang, D Ma, and D Cai. With greater text comes greater necessity: Inference-time training helps long text generation. *arXiv preprint arXiv:2401.11504*, 2024m.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, et al. Memoryllm: Towards self-updatable large language models. *arXiv preprint arXiv:2402.04624*, 2024n.

Yu Wang, Zeyuan Zhang, Julian McAuley, and Zexue He. Lvchat: Facilitating long video comprehension. *arXiv preprint arXiv:2402.12079*, 2024o.

Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International conference on machine learning*, pp. 5123–5132. PMLR, 2018.

Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9154–9162, 2019.

Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, S Yu Philip, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2208–2225, 2022.

Yuxuan Wang, Yueqian Wang, Pengfei Wu, Jianxin Liang, Dongyan Zhao, Yang Liu, and Zilong Zheng. Efficient temporal extrapolation of multimodal large language models with temporal grounding bridge. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9972–9987, 2024p.

Yuxuan Wang, Cihang Xie, Yang Liu, and Zilong Zheng. Videollamb: Long-context video understanding with recurrent memory bridges. *arXiv preprint arXiv:2409.01071*, 2024q.

Zheng Wang, Boxiao Jin, Zhongzhi Yu, and Minjia Zhang. Model tells you where to merge: Adaptive kv cache merging for llms on long-context tasks. *arXiv preprint arXiv:2407.08454*, 2024r.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv preprint arXiv:2405.19209*, 2024s.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. In *European Conference on Computer Vision*, pp. 453–470. Springer, 2025.

Bingyang Wu, Shengyu Liu, Yinmin Zhong, Peng Sun, Xuanzhe Liu, and Xin Jin. Loongserve: Efficiently serving long-context large language models with elastic sequence parallelism. In *Proceedings of the ACM SIGOPS 30th Symposium on Operating Systems Principles*, pp. 640–654, 2024a.

Chan Wu, Hanxiao Zhang, Lin Ju, Jinjing Huang, Youshao Xiao, Zhaoxin Huan, Siyuan Li, Fanzhuang Meng, Lei Liang, Xiaolu Zhang, et al. Rethinking memory and communication cost for efficient large language model training. *arXiv preprint arXiv:2310.06003*, 2023.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b.

Haoyi Wu and Kewei Tu. Layer-condensed kv cache for efficient inference of large language models. *arXiv preprint arXiv:2405.10637*, 2024.

Jialong Wu, Zhenglin Wang, Linhai Zhang, Yilong Lai, Yulan He, and Deyu Zhou. Scope: Optimizing key-value cache compression in long-context generation. *arXiv preprint arXiv:2412.13649*, 2024c.

Liangxuan Wu, Yanjie Zhao, Chao Wang, Tianming Liu, and Haoyu Wang. A first look at llm-powered smartphones. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering Workshops*, pp. 208–217, 2024d.

Mengdi Wu, Xinhao Cheng, Oded Padon, and Zhihao Jia. A multi-level superoptimizer for tensor programs. *arXiv preprint arXiv:2405.05751*, 2024e.

Tong Wu, Yanpeng Zhao, and Zilong Zheng. An efficient recipe for long context extension via middle-focused positional encoding. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024f.

Wei Wu, Kecheng Zheng, Shuailei Ma, Fan Lu, Yuxin Guo, Yifei Zhang, Wei Chen, Qingpei Guo, Yujun Shen, and Zheng-Jun Zha. Lotlip: Improving language-image pre-training for long text understanding. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024g.

Wenhao Wu. Freeva: Offline mllm as training-free video assistant. *arXiv preprint arXiv:2405.07798*, 2024.

Wenhao Wu, Yizhong Wang, Yao Fu, Xiang Yue, Dawei Zhu, and Sujian Li. Long context alignment with short instructions and synthesized positions. *arXiv preprint arXiv:2405.03939*, 2024h.

Xiaodong Wu, Minhao Wang, Yichen Liu, Xiaoming Shi, He Yan, Xiangju Lu, Junmin Zhu, and Wei Zhang. Lifbench: Evaluating the instruction following performance and stability of large language models in long-context scenarios. *arXiv preprint arXiv:2411.07037*, 2024i.

Yingsheng Wu, Yuxuan Gu, Xiaocheng Feng, Weihong Zhong, Dongliang Xu, Qing Yang, Hongtao Liu, and Bing Qin. Extending context window of large language models from a distributional perspective. *arXiv preprint arXiv:2410.01490*, 2024j.

Yuhao Wu, Ming Shan Hee, Zhiqing Hu, and Roy Ka-Wei Lee. Longgenbench: Benchmarking long-form generation in long context llms. *arXiv preprint arXiv:2409.02076*, 2024k.

Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers. *arXiv preprint arXiv:2203.08913*, 2022.

Haocheng Xi, Changhao Li, Jianfei Chen, and Jun Zhu. Training transformers with 4-bit integers. *Advances in Neural Information Processing Systems*, 36:49146–49168, 2023.

Haocheng Xi, Yuxiang Chen, Kang Zhao, Kai Jun Teh, Jianfei Chen, and Jun Zhu. Jetfire: Efficient and accurate transformer pretraining with int8 data flow and per-block quantization. *arXiv preprint arXiv:2403.12422*, 2024.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. Infllm: Training-free long-context extrapolation for llms with an efficient context memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. *arXiv preprint arXiv:2410.10819*, 2024b.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024c.

Xudong Xie, Liang Yin, Hao Yan, Yang Liu, Jing Ding, Minghui Liao, Yuliang Liu, Wei Chen, and Xiang Bai. Wukong: A large multimodal model for efficient long pdf reading with end-to-end sparse sampling. *arXiv preprint arXiv:2410.05970*, 2024.

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12):1486–1496, 2023.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. Effective long-context scaling of foundation models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4643–4663, 2024a.

Zheyang Xiong, Vasilis Papageorgiou, Kangwook Lee, and Dimitris Papailiopoulos. From artificial needles to real haystacks: Improving retrieval capabilities in llms by finetuning on synthetic data, 2024b. URL https://arxiv.org/abs/2406.19292.

Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. On-device language models: A comprehensive review. *arXiv preprint arXiv:2409.00088*, 2024a.

Jiale Xu, Rui Zhang, Cong Guo, Weiming Hu, Zihan Liu, Feiyang Wu, Yu Feng, Shixuan Sun, Changxu Shao, Yuhong Guo, et al. vtensor: Flexible virtual tensor management for efficient llm serving. *arXiv preprint arXiv:2407.15309*, 2024b.

Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024c.

Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024d.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*, 2023.

Xin Xu and Zhouchen Lin. Mixcon: A hybrid architecture for efficient and adaptive sequence modeling. In *ECAI 2024*, pp. 1027–1034. IOS Press, 2024.

Yuhui Xu, Zhanming Jie, Hanze Dong, Lei Wang, Xudong Lu, Aojun Zhou, Amrita Saha, Caiming Xiong, and Doyen Sahoo. Think: Thinner key cache by query-driven pruning. *arXiv preprint arXiv:2407.21018*, 2024e.

Zhe Xu, Jiasheng Ye, Xiangyang Liu, Tianxiang Sun, Xiaoran Liu, Qipeng Guo, Linlin Li, Qun Liu, Xuanjing Huang, and Xipeng Qiu. Detectiveqa: Evaluating long-context reasoning on detective novels. *arXiv preprint arXiv:2409.02465*, 2024f.

Chunyu Xue, Weihao Cui, Han Zhao, Quan Chen, Shulai Zhang, Pengyu Yang, Jing Yang, Shaobo Li, and Minyi Guo. A codesign of scheduling and parallelization for large model training in heterogeneous clusters. *arXiv preprint arXiv:2403.16125*, 2024a.

Fuzhao Xue, Yukang Chen, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024b.

Zhenliang Xue, Yixin Song, Zeyu Mi, Le Chen, Yubin Xia, and Haibo Chen. Powerinfer-2: Fast large language model inference on a smartphone. *arXiv preprint arXiv:2406.06282*, 2024c.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. Tener: adapting transformer encoder for named entity recognition. *arXiv preprint arXiv:1911.04474*, 2019.

Ruiqing Yan, Linghan Zheng, Xingbo Du, Han Zou, Yufeng Guo, and Jianfei Yang. Recurformer: Not all transformer heads need self-attention. *arXiv preprint arXiv:2410.12850*, 2024.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a. URL https://arxiv.org/abs/2407.10671.

Dongjie Yang, XiaoDong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao. Pyramidinfer: Pyramid kv cache compression for high-throughput llm inference. *arXiv preprint arXiv:2405.12532*, 2024b.

Hongkang Yang, Zehao Lin, Wenjin Wang, Hao Wu, Zhiyu Li, Bo Tang, Wenqiang Wei, Jinbo Wang, Zeyun Tang, Shichao Song, et al. Memory3: Language modeling with explicit memory. *arXiv preprint arXiv:2407.01178*, 2024c.

Hua Yang, Duohai Li, and Shiman Li. Mcsd: An efficient language model with diverse fusion. *arXiv preprint arXiv:2406.12230*, 2024d.

Jingyi Yang, Aya Ibrahim, Xinfeng Xie, Bangsheng Tang, Grigory Sizov, Jongsoo Park, Jianyu Huang, et al. Context parallelism for scalable million-token inference. *arXiv preprint arXiv:2411.01783*, 2024e.

June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. No token left behind: Reliable kv cache compression via importance-aware mixed precision quantization. *arXiv preprint arXiv:2402.18096*, 2024f.

Kai Yang, Jan Ackermann, Zhenyu He, Guhao Feng, Bohang Zhang, Yunzhen Feng, Qiwei Ye, Di He, and Liwei Wang. Do efficient transformers really save computation? *arXiv preprint arXiv:2402.13934*, 2024g.

Shuai Yang, Yuying Ge, Yang Li, Yukang Chen, Yixiao Ge, Ying Shan, and Yingcong Chen. Seed-story: Multimodal long story generation with large language model. *arXiv preprint arXiv:2407.08683*, 2024h.

Shuo Yang, Ying Sheng, Joseph E Gonzalez, Ion Stoica, and Lianmin Zheng. Post-training sparse attention with double sparsity. *arXiv preprint arXiv:2408.07092*, 2024i.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Forty-first International Conference on Machine Learning*, 2024j.

Yifei Yang, Zouying Cao, Qiguang Chen, Libo Qin, Dongjie Yang, Hai Zhao, and Zhi Chen. Kvsharer: Efficient inference via layer-wise dissimilar kv cache sharing. *arXiv preprint arXiv:2410.18517*, 2024k.

Zhen Yang, JN Han, Kan Wu, Ruobing Xie, An Wang, Xingwu Sun, and Zhanhui Kang. Lossless kv cache compression to 2%. *arXiv preprint arXiv:2410.15252*, 2024l.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. Cacheblend: Fast large language model serving with cached knowledge fusion. *arXiv preprint arXiv:2405.16444*, 2024a.

Jinghan Yao, Sam Ade Jacobs, Masahiro Tanaka, Olatunji Ruwase, Aamir Shafi, Hari Subramoni, and Dhabaleswar K Panda. Training ultra long context language model with fully pipelined distributed transformer. *arXiv preprint arXiv:2408.16978*, 2024b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024c.

Yao Yao, Zuchao Li, and Hai Zhao. Sirllm: Streaming infinite retentive llm. *arXiv preprint arXiv:2405.12528*, 2024d.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024e.

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024a.

Xi Ye, Fangcong Yin, Yinghui He, Joie Zhang, Howard Yen, Tianyu Gao, Greg Durrett, and Danqi Chen. Longproc: Benchmarking long-context language models on long procedural generation. *arXiv preprint arXiv:2501.05414*, 2025a.

Zihao Ye, Lequn Chen, Ruihang Lai, Yilong Zhao, Size Zheng, Junru Shao, Bohan Hou, Hongyi Jin, Yifei Zuo, Liangsheng Yin, Tianqi Chen, and Luis Ceze. Accelerating self-attentions for llm serving with flashinfer, February 2024b. URL https://flashinfer.ai/2024/02/02/introduce-flashinfer.html.

Zihao Ye, Ruihang Lai, Bo-Ru Lu, Chien-Yu Lin, Size Zheng, Lequn Chen, Tianqi Chen, and Luis Ceze. Cascade inference: Memory bandwidth efficient shared prefix batch decoding, February 2024c. URL https://flashinfer.ai/2024/02/02/cascade-inference.html.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, et al. Flashinfer: Efficient and customizable attention engine for llm inference serving. *arXiv preprint arXiv:2501.01005*, 2025b.

Howard Yen, Tianyu Gao, and Danqi Chen. Long-context language modeling with parallel context encoding. *arXiv preprint arXiv:2402.16617*, 2024a.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. Helmet: How to evaluate long-context language models effectively and thoroughly. *arXiv preprint arXiv:2410.02694*, 2024b.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Yunhang Shen, Chunjiang Ge, Yan Yang, Zuwei Long, Yuhan Dai, Tong Xu, Xing Sun, et al. T2vid: Translating long text into multi-image is the catalyst for video-llms. *arXiv preprint arXiv:2411.19951*, 2024a.

Wangsong Yin, Mengwei Xu, Yuanchun Li, and Xuanzhe Liu. Llm as a system service on mobile devices. *arXiv preprint arXiv:2403.11805*, 2024b.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Qinyuan Cheng, Xipeng Qiu, and Xuan-Jing Huang. Explicit memory learning with expectation maximization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16618–16635, 2024c.

Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. Compact: Compressing retrieved documents actively for question answering. *arXiv preprint arXiv:2407.09014*, 2024.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. Orca: A distributed serving system for Transformer-Based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pp. 521–538, Carlsbad, CA, July 2022. USENIX Association. ISBN 978-1-939133-28-1. URL https://www.usenix.org/conference/osdi22/presentation/yu.

Hao Yu, Zelan Yang, Shen Li, Yong Li, and Jianxin Wu. Effectively compress kv heads for llm. *arXiv preprint arXiv:2406.07056*, 2024a.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, et al. Frame-voyager: Learning to query frames for video large language models. *arXiv preprint arXiv:2410.03226*, 2024b.

Yijiong Yu, Yongfeng Huang, Zhixiao Qi, and Zhe Zhou. Training with "paraphrasing the original text" teaches llm to better retrieve in long-context tasks. *arXiv preprint arXiv:2312.11193*, 2023.

Yijiong Yu, Ma Xiufa, Fang Jianwei, Zhi Xu, Su Guangyao, Wang Jiancheng, Yongfeng Huang, Zhixiao Qi, Wei Wang, Weifeng Liu, et al. Hyper-multi-step: The truth behind difficult long-context tasks. *arXiv preprint arXiv:2410.04422*, 2024c.

Danlong Yuan, Jiahao Liu, Bei Li, Huishuai Zhang, Jingang Wang, Xunliang Cai, and Dongyan Zhao. Remamba: Equip mamba with effective long-sequence modeling. *arXiv preprint arXiv:2408.15496*, 2024a.

Tao Yuan, Xuefei Ning, Dong Zhou, Zhijie Yang, Shiyao Li, Minghui Zhuang, Zheyue Tan, Zhuyu Yao, Dahua Lin, Boxun Li, et al. Lv-eval: A balanced long-context benchmark with 5 length levels up to 256k. *arXiv preprint arXiv:2402.05136*, 2024b.

Zhihang Yuan, Yuzhang Shang, Yang Zhou, Zhen Dong, Zhe Zhou, Chenhao Xue, Bingzhe Wu, Zhikai Li, Qingyi Gu, Yong Jae Lee, et al. Llm inference unveiled: Survey and roofline model insights. *arXiv preprint arXiv:2402.16363*, 2024c.

Amir Zandieh, Majid Daliri, and Insu Han. Qjl: 1-bit quantized jl transform for kv cache quantization with zero overhead. *arXiv preprint arXiv:2406.03482*, 2024.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Bo Wang, Shimin Li, Yunhua Zhou, Qipeng Guo, Xuanjing Huang, and Xipeng Qiu. Scaling of search and learning: A roadmap to reproduce o1 from reinforcement learning perspective. *arXiv preprint arXiv:2412.14135*, 2024a.

Zhiyuan Zeng, Qipeng Guo, Xiaoran Liu, Zhangyue Yin, Wentao Shu, Mianqiu Huang, Bo Wang, Yunhua Zhou, Linlin Li, Qun Liu, et al. Memorize step by step: Efficient long-context prefilling with incremental memory and decremental chunk. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21021–21034, 2024b.

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv preprint arXiv:2402.12226*, 2024.

Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. In *European Conference on Computer Vision*, pp. 310–325. Springer, 2025.

Bowen Zhang, Hexiang Hu, Linlu Qiu, Peter Shaw, and Fei Sha. Visually grounded concept composition. *arXiv preprint arXiv:2109.14115*, 2021.

Caiqi Zhang, Fangyu Liu, Marco Basaldella, and Nigel Collier. Luq: Long-text uncertainty quantification for llms. *arXiv preprint arXiv:2403.20279*, 2024a.

Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering. *arXiv preprint arXiv:2312.17235*, 2023a.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15757–15773, 2023b.

Dong Zhang, Xin Zhang, Jun Zhan, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speechgpt-gen: Scaling chain-of-information speech generation. *arXiv preprint arXiv:2401.13527*, 2024b.

Hailin Zhang, Xiaodong Ji, Yilin Chen, Fangcheng Fu, Xupeng Miao, Xiaonan Nie, Weipeng Chen, and Bin Cui. Pqcache: Product quantization-based kvcache for long context llm inference. *arXiv preprint arXiv:2407.12820*, 2024c.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 543–553, 2023c.

Hongjie Zhang, Yi Liu, Lu Dong, Yifei Huang, Zhen-Hua Ling, Yali Wang, Limin Wang, and Yu Qiao. Movqa: A benchmark of versatile question-answering for long-form movie understanding. *arXiv preprint arXiv:2312.04817*, 2023d.

Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, et al. Longcite: Enabling llms to generate fine-grained citations in long-context qa. *arXiv preprint arXiv:2409.02897*, 2024d.

Jiajie Zhang, Zhongni Hou, Xin Lv, Shulin Cao, Zhenyu Hou, Yilin Niu, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. Longreward: Improving long-context large language models with ai feedback. *arXiv preprint arXiv:2410.21252*, 2024e.

Jianhao Zhang, Shihan Ma, Peihong Liu, and Jinhui Yuan. Coop: Memory is not a commodity. *Advances in Neural Information Processing Systems*, 36, 2024f.

Kechi Zhang, Ge Li, Huangzhao Zhang, and Zhi Jin. Hirope: Length extrapolation for code models using hierarchical position. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13615–13627, 2024g.

Lei Zhang, Yunshui Li, Ziqiang Liu, Junhao Liu, Longze Chen, Run Luo, Min Yang, et al. Marathon: A race through the realm of long context with large language models. *arXiv preprint arXiv:2312.09542*, 2023e.

Meiru Zhang, Zaiqiao Meng, and Nigel Collier. Attention instruction: Amplifying attention in the middle via prompting. *arXiv preprint arXiv:2406.17095*, 2024h.

Mozhi Zhang, Pengyu Wang, Chenkun Tan, Mianqiu Huang, Dong Zhang, Yaqian Zhou, and Xipeng Qiu. Metaalign: Align large language models with diverse preferences during inference time. *arXiv preprint arXiv:2410.14184*, 2024i.

Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, Lin Chen, Qipeng Guo, et al. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*, 2024j.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. *arXiv preprint arXiv:2401.03462*, 2024k.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *URL https://arxiv.org/abs/2406.16852*, 2024l.

Rongzhi Zhang, Kuang Wang, Liyuan Liu, Shuohang Wang, Hao Cheng, Chao Zhang, and Yelong Shen. Lorc: Low-rank compression for llms kv cache with a progressive compression strategy. *arXiv preprint arXiv:2410.03111*, 2024m.

Ruisi Zhang, Tianyu Liu, Will Feng, Andrew Gu, Sanket Purandare, Wanchao Liang, and Francisco Massa. Simplefsdp: Simpler fully sharded data parallel with torch.compile. *ArXiv*, abs/2411.00284, 2024n. URL https://api.semanticscholar.org/CorpusID:273798121.

Xiang Zhang, Senyu Li, Ning Shi, Bradley Hauer, Zijun Wu, Grzegorz Kondrak, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. Cross-modal consistency in multimodal large language models. *arXiv preprint arXiv:2411.09273*, 2024o.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, et al. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1393–1412, 2024p.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, et al. Infinitebench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15262–15277, 2024q.

Xuan Zhang, Cunxiao Du, Chao Du, Tianyu Pang, Wei Gao, and Min Lin. Simlayerkv: A simple framework for layer-level kv cache reduction. *arXiv preprint arXiv:2410.13846*, 2024r.

Yikai Zhang, Junlong Li, and Pengfei Liu. Extending llms' context window with 100 samples. *arXiv preprint arXiv:2401.07004*, 2024s.

Yu Zhang, Songlin Yang, Rui-Jie Zhu, Yue Zhang, Leyang Cui, Yiqiao Wang, Bolun Wang, Freda Shi, Bailin Wang, Wei Bi, et al. Gated slot attention for efficient linear-time sequence modeling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024t.

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, April 2024u. URL https://llava-vl.github.io/blog/2024-04-30-llava-next-video/.

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. Cam: Cache merging for memory-efficient llms inference. In *Forty-first International Conference on Machine Learning*, 2024v.

Zhen Zhang, Shuai Zheng, Yida Wang, Justin Chiu, George Karypis, Trishul M. Chilimbi, Mu Li, and Xin Jin. Mics: Near-linear scaling for training gigantic model on public cloud. *Proc. VLDB Endow.*, 16:37–50, 2022.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023f.

Hanyu Zhao, Zhenhua Han, Zhi Yang, Quanlu Zhang, Mingxia Li, Fan Yang, Qianxi Zhang, Binyang Li, Yuqing Yang, Lili Qiu, et al. Silod: A co-design of caching and scheduling for deep learning clusters. In *Proceedings of the Eighteenth European Conference on Computer Systems*, pp. 883–898, 2023a.

Jun Zhao, Can Zu, Hao Xu, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. Longagent: Scaling language models to 128k context through multi-agent collaboration. *arXiv preprint arXiv:2402.11550*, 2024a.

Liang Zhao, Xiaocheng Feng, Xiachong Feng, Bin Qin, and Ting Liu. Length extrapolation of transformers: A survey from the perspective of position encoding. *arXiv preprint arXiv:2312.17044*, 2023b.

Liang Zhao, Tianwen Wei, Liang Zeng, Cheng Cheng, Liu Yang, Peng Cheng, Lijie Wang, Chenxia Li, Xuejie Wu, Bo Zhu, et al. Longskywork: A training recipe for efficiently extending context length in large language models. *arXiv preprint arXiv:2406.00605*, 2024b.

Pinxue Zhao, Hailin Zhang, Fangcheng Fu, Xiaonan Nie, Qibin Liu, Fang Yang, Yuanbo Peng, Dian Jiao, Shuaipeng Li, Jinbao Xue, Yangyu Tao, and Bin Cui. Efficiently training 7b llm with 1 million sequence length on 8 gpus. *arXiv preprint arXiv:2407.12117*, 2024c.

Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, Yuxiao Dong, and Jie Tang. Longrag: A dual-perspective retrieval-augmented generation paradigm for long-context question answering. *arXiv preprint arXiv:2410.18050*, 2024d.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023c.

Yu Zhao, Yuanbin Qu, Konrad Staniszewski, Szymon Tworkowski, Wei Liu, Piotr Miłoś, Yuxiang Wu, and Pasquale Minervini. Analysing the impact of sequence composition on language model pre-training. *arXiv preprint arXiv:2402.13991*, 2024e.

Chuanyang Zheng, Yihang Gao, Han Shi, Minbin Huang, Jingyao Li, Jing Xiong, Xiaozhe Ren, Michael Ng, Xin Jiang, Zhenguo Li, et al. Dape: Data-adaptive positional encoding for length extrapolation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Chuanyang Zheng, Yihang Gao, Han Shi, Jing Xiong, Jiankai Sun, Jingyao Li, Minbin Huang, Xiaozhe Ren, Michael Ng, Xin Jiang, et al. Dape v2: Process attention score as feature map for length extrapolation. *arXiv preprint arXiv:2410.04798*, 2024b.

Kecheng Zheng, Yifei Zhang, Wei Wu, Fan Lu, Shuailei Ma, Xin Jin, Wei Chen, and Yujun Shen. Dreamlip: Language-image pre-training with long captions. In *European Conference on Computer Vision*, pp. 73–90. Springer, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *arXiv preprint arXiv:2312.07104*, 2024c.

Zhihu & ModelBest Inc. Zhilight, 2024. URL https://github.com/zhihu/ZhiLight.

Meizhi Zhong, Chen Zhang, Yikun Lei, Xikai Liu, Yan Gao, Yao Hu, Kehai Chen, and Min Zhang. Understanding the rope extensions of long-context llms: An attention perspective. *arXiv preprint arXiv:2406.13282*, 2024a.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. Qmsum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5905–5921, 2021.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024b.

Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. {DistServe}: Disaggregating prefill and decoding for goodput-optimized large language model serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pp. 193–210, 2024c.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024a.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Peng Cui, Tiannan Wang, Zhenxin Xiao, Yifan Hou, Ryan Cotterell, and Mrinmaya Sachan. Recurrentgpt: Interactive generation of (arbitrarily) long text. *arXiv preprint arXiv:2305.13304*, 2023.

Zihan Zhou, Chong Li, Xinyi Chen, Shuo Wang, Yu Chao, Zhili Li, Haoyu Wang, Rongqiao An, Qi Shi, Zhixing Tan, et al. Llmxmapreduce: Simplified long-sequence processing using large language models. *arXiv preprint arXiv:2410.09342*, 2024b.

Dawei Zhu, Nan Yang, Liang Wang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. Pose: Efficient context window extension of llms via positional skip-wise training. *arXiv preprint arXiv:2309.10400*, 2023.

Qianchao Zhu, Jiangfei Duan, Chang Chen, Siran Liu, Xiuhong Li, Guanyu Feng, Xin Lv, Huanqi Cao, Xiao Chuanfu, Xingcheng Zhang, et al. Sampleattention: Near-lossless acceleration of long context llm inference with adaptive structured sparse attention. *arXiv preprint arXiv:2406.15486*, 2024.

Heqing Zou, Tianze Luo, Guiyang Xie, Fengmao Lv, Guangcong Wang, Juanyang Chen, Zhuochen Wang, Hansheng Zhang, Huaijian Zhang, et al. From seconds to hours: Reviewing multimodal large language models on comprehensive long video understanding. *arXiv preprint arXiv:2409.18938*, 2024a.

Kaijian Zou, Muhammad Khalifa, and Lu Wang. Retrieval or global context understanding? on many-shot in-context learning for long-context evaluation. *arXiv preprint arXiv:2411.07130*, 2024b.

Zayd Muhammad Kawakibi Zuhri, Muhammad Farid Adilazuarda, Ayu Purwarianti, and Alham Fikri Aji. Mlkv: Multi-layer key-value heads for memory efficient transformer decoding. *arXiv preprint arXiv:2406.09297*, 2024.

Jingwei Zuo, Maksim Velikanov, Dhia Eddine Rhaiem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. Falcon mamba: The first competitive attention-free 7b language model. *arXiv preprint arXiv:2410.05355*, 2024.