When Linear Attention Meets Autoregressive Decoding: Towards More Effective and Efficient Linearized Large Language Models

Haoran You [1] **Yichao Fu** [1] **Zheng Wang** [1] **Amir Yazdanbakhsh** [2] **Yingyan (Celine) Lin** [1]

Abstract

Autoregressive Large Language Models (LLMs) have achieved impressive performance in language tasks but face significant bottlenecks: (1) quadratic complexity bottleneck in the attention module with increasing token numbers, and (2) efficiency bottleneck due to the sequential processing nature of autoregressive LLMs during generation. Linear attention and speculative decoding emerge as solutions for these challenges, yet their applicability and combinatory potential for autoregressive LLMs remain uncertain. To this end, we embark on the first comprehensive empirical investigation into the efficacy of existing linear attention methods for autoregressive LLMs and their integration with speculative decoding. We introduce an augmentation technique for linear attention and ensure the compatibility between linear attention and speculative decoding for efficient LLM training and serving. Extensive experiments and ablation studies on seven existing linear attention works and five encoder/decoder-based LLMs consistently validate the effectiveness of our augmented linearized LLMs, e.g., achieving up to a 6.67 perplexity reduction on LLaMA and 2× speedups during generation as compared to prior linear attention methods.

1. Introduction

LLMs have showcased exceptional capabilities in language understanding and generation tasks, sparking significant and widespread excitement. Among these, autoregressive LLMs such as OpenAI's ChatGPT (OpenAI, 2023a;b), Meta's LLaMA (Touvron et al., 2023a;b), and Google's Bard (Waisberg et al., 2023) stand out due to their state-of-the-art (SOTA) generation abilities. However, the remarkable performance of autoregressive LLMs is compromised by significant computational and memory demands, attributed to two primary bottlenecks that hinder efficient training and serving. ***Bottleneck 1***: There is an inherent complexity bottleneck within LLMs, the core attention module exhibits quadratic complexity relative to the length of the input sequence. Hence, LLMs are often trained with a pre-defined relatively short context size, such as 2048 tokens for LLaMA, restricting their application in tasks like summarizing lengthy documents or responding to extensive questions (Chen et al., 2023b). ***Bottleneck 2***: The sequential processing nature of autoregressive LLMs introduces unique efficiency bottlenecks, particularly demonstrated in their low parallelism during serving or generation phases (Miao et al., 2023). These two bottlenecks collectively impede the models' efficiency and broader application potential.

To address the aforementioned bottlenecks and fully unlock the potential of LLMs, various techniques have been developed, such as pruning (Ma et al., 2023), quantization (Frantar et al., 2022; Xiao et al., 2023), speculative decoding (Miao et al., 2023; Leviathan et al., 2023), and linear attention (Qin et al., 2023; Lu et al., 2021). Among these methods, linear attention stands out as an effective way to address the ***Bottleneck 1*** by replacing the quadratic complexity of the original softmax attention with linear complexity. Speculative decoding has the potential to tackle the ***Bottleneck 2*** by enhancing token-level parallelism via initially using smaller draft models for speculative decoding, followed by larger LLMs for verification (Miao et al., 2023; Cai et al., 2023b; Chen et al., 2023a). All other techniques do not change the attention mechanism nor token-level parallelism patterns and are thus orthogonal methods.

However, there are still two unclear questions. ***Q1***: whether existing linear attention methods, which are mostly designed for encoder-based LLMs like BERT (Devlin et al., 2018) or Vision Transformers (ViTs) (Dosovitskiy et al., 2021), still apply to autoregressive decoder-based LLMs. ***Q2***: whether linear attention and speculative decoding can be effectively combined to collectively address the two bottlenecks during LLM training and serving. To this end, this paper undertakes the first comprehensive empirical exploration to assess the effectiveness of linearized autoregressive LLMs and

[1]School of Computer Science, Georgia Institute of Technology, Atlanta, USA [2]Google DeepMind, San Jose, USA. Correspondence to: Haoran You <haoran.you@gatech.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

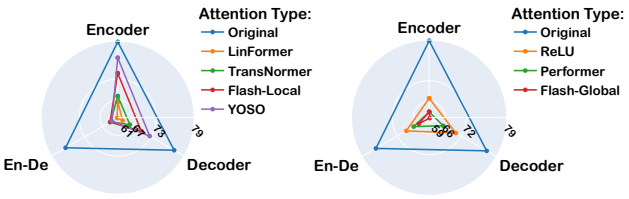

Figure 1: Empirical examination of seven linear attention methods on top of three categories of LLMs evaluated on GLUE (Wang et al., 2018): (1) encoder-based BERT (Devlin et al., 2018); (2) decoder-based GPT-2 (Radford et al., 2019); and (3) encoder-decoder T5 (Roberts et al., 2022). **Left:** The majority of SOTA linear attentions (Wang et al., 2020; Katharopoulos et al., 2020; Zeng et al., 2021; Hua et al., 2022) exhibit superior performance on encoder-based models compared to decoder-based ones. **Right:** Other linear attention works (Choromanski et al., 2021; Cai et al., 2023a) consistently perform less effectively on all LLMs.

the compatibility between linear attention and speculative decoding. For ***Q1***, our findings reveal that existing linear attention methods exhibit reduced efficacy in autoregressive LLMs. This is because autoregressive LLMs require the ability to handle temporal dependencies for predicting future generations. In these LLMs, attention maps are applied with a lower-triangular mask to accurately represent temporal-dependent information. Hence, directly adopting existing linear attention methods can result in several complications. For example, linear attention augmentation methods (You et al., 2023; Xiong et al., 2021) employ efficient depthwise convolution to enhance local information capture. Despite this improvement, such convolution leads to information leakage, as it involves contexts from future time steps during training. For ***Q2***, our findings show that directly combining the linear attention and the speculative decoding methods does not work effectively. This is because large LLMs in speculative decoding have to employ tree-based attention, i.e., masked attention which allows simultaneous processing of multiple candidates while restricting each token's access to its antecedent tokens. While existing linear attention designs do not consider adaptively replicating the temporal dependencies introduced by speculative decoding on the fly. The mismatch between them poses a significant challenge in optimizing the trade-off between accuracy and efficiency.

Motivated by the aforementioned challenges, we propose an effective local convolutional augmentation to prevent information leakage, enhance performance, and maintain compatibility with speculative decoding. Specifically, our contributions are summarized as follows:

- We conduct a comprehensive evaluation of seven linear attention methods across three types of LLMs, revealing that most existing linear attentions, initially designed for encoder-based LLMs or ViTs, are not optimally suited for autoregressive decoder-based LLMs.
- We examine the limitations of current linear attention methods and introduce an ***effective*** local augmentation technique. This technique is designed to enhance the local feature extraction ability of autoregressive LLMs without causing any information leakage.
- We develop a solution for the integration of linear attention with speculative decoding's tree-based attention, aiming to boost token-level parallelism for ***efficient*** generation and seeking to accelerate not just the training but also the serving phase of LLMs.
- We conduct extensive experiments to evaluate the effectiveness of our augmented linearized LLMs. Results on five LLMs consistently demonstrate enhanced accuracies (up to 6.67 perplexity reduction) achieved by local augmentation and 2× speedups during generation thanks to our integration with speculative decoding.

2. Related Works

Autoregressive LLMs. Transformers (Vaswani et al., 2017; Dosovitskiy et al., 2021) have significantly advanced the fields of language and vision, leading to the development of foundation LLMs such as ChatGPT (Brown et al., 2020; OpenAI, 2023b), LLaMA (Touvron et al., 2023a;b), Bard (Waisberg et al., 2023), DALL-E (Ramesh et al., 2021), etc. These models fall into three main categories: *encoder-based*, *decoder-based*, and *encoder-decoder* models. Encoder-based models like BERT (Devlin et al., 2018) focus on natural language understanding and are also commonly used in image processing (Dosovitskiy et al., 2021). Encoder-decoder models like the original Transformer (Vaswani et al., 2017), Bard (Waisberg et al., 2023), and T5 (Raffel et al., 2020; Roberts et al., 2022) are designed for sequence-to-sequence tasks (e.g., translation, speech recognition), where the encoder extracts features and the decoder produces outputs based on these features. Decoder-based models, including GPT (Radford et al., 2019; OpenAI, 2023b) and LLaMA (Touvron et al., 2023a), generate text sequentially by predicting the next token based on previous ones. All these models leverage Transformer architectures but differ in their specific purposes and structures. Our work presents a comprehensive study of applying linear attention techniques to the encoder/decoder-based LLMs.

Efficient Linear Attention Transformers' self-attention modules, known for their quadratic computational complexity (Zhu et al., 2021; Katharopoulos et al., 2020), have spurred the development of linear attention methods to improve efficiency, especially in encoder-based LLMs for better training and inference. Techniques such as local attentions (Liu et al., 2021; Arar et al., 2022; Wang et al., 2020; Tu et al., 2022) limit self-attention to neighboring tokens or group attention queries to reduce computational cost, while kernel-based linear attentions (Liu et al., 2021; Arar et al., 2022; Wang et al., 2020; Tu et al., 2022) decompose the softmax with kernel functions and exchange the

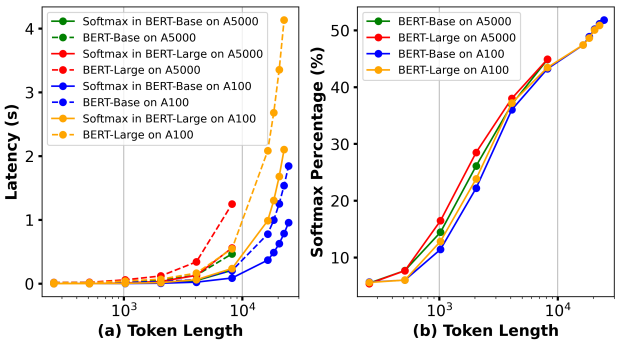

Figure 2: Runtime profiling: **(a)** actual runtime latencies for both the softmax and the entire model; **(b)** the percentage of time allocated to softmax computations across the latency of the entire model. We collect all data using BERT-Base/Large models on a single A5000 or A100 GPU.

computation order. However, only a few linear attention approaches focus on decoder-based autoregressive LLMs, aiming to reduce RNN-style sequential state updates over a large number of steps (Hua et al., 2022; Katharopoulos et al., 2020). Recent studies, like LongLoRA (Chen et al., 2023b), aim to adapt local attention techniques for efficient fine-tuning of pre-trained autoregressive LLMs, yet a thorough analysis comparing various linear attention methods for autoregressive LLMs remains lacking. This paper uniquely provides a systematic review of existing linear attentions for decoder-based autoregressive LLMs and investigates how to efficiently enhance less effective linear attention methods.

Speculative Decoding. Linear attention methods reduce training inefficiencies. Yet, autoregressive decoding's sequential nature limits parallelism during deployment, constraining input token numbers. Speculative decoding (Chen et al., 2023a; Miao et al., 2023; Kim et al., 2023; Leviathan et al., 2023; Cai et al., 2023b) has proven to be an effective strategy for boosting parallelism in LLM serving, utilizing small speculative models for initial generation, with original LLMs serving as validators to assess if the output meets standards or needs resampling. Recent works like Medusa (Cai et al., 2023b) further argue that the small speculative models and LLMs can be the same model, and other studies (Schuster et al., 2022; Bae et al., 2023) suggest using shallow layers for generation and deeper layers for verification, based on early exit strategies. Such speculative decoding and linear attention jointly ensure efficient LLM training and generation, especially for long sequence inputs. In this paper, we take the initiative to investigate the synergy between linearized LLMs and speculative sampling, to improve the efficiency of training and serving LLMs.

3. Preliminaries and Evaluation

In this section, we recap the linear attention and autoregressive LLMs and conduct a comprehensive evaluation of seven linear attention methods across three LLM types.

3.1. Preliminaries of Linear Attention and LLMs

Self-Attention and Softmax Bottleneck. Self-attention module is a core component of the Transformer (Vaswani et al., 2017; Dosovitskiy et al., 2021), and typically includes multiple heads. Each head computes global-context information by evaluating pairwise correlations among all n tokens (n represents the total number of tokens) as follows:

$$\begin{aligned} &\texttt{Attn}(\mathbf{X}) = \texttt{Concat}(\mathrm{H}_1, \cdots, \mathrm{H}_h) \cdot \mathbf{W}_O, \text{ where} \\ &\mathrm{H}_i = \texttt{Softmax}\left(\frac{f_Q(\mathbf{X}) \cdot f_K(\mathbf{X})^T}{\sqrt{d_k}}\right) \cdot f_V(\mathbf{X}), \end{aligned} \tag{1}$$

where h denotes the number of heads. Within each head H_i, input tokens $\mathbf{X} \in \mathbb{R}^{n \times d}$ of length n and dimension d will be linearly projected to query, key, and value matrices, i.e., $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{n \times d_k}$, through three linear mapping functions, $f_Q = \mathbf{X}\mathbf{W}_Q, f_K = \mathbf{X}\mathbf{W}_K, f_V = \mathbf{X}\mathbf{W}_V$, where $d_k = d/h$ is the embedding dimension of each head and $\mathbf{W}_{Q/K/V}$ are the associated weight matrices. The final outputs are generated by concatenating the results from all heads and applying a weight matrix $\mathbf{W}_O \in \mathbb{R}^{d \times d}$.

Within self-attention, it is observed that the softmax becomes a memory bottleneck when dealing with long sequences (Dao et al., 2022). The substantial number of memory accesses leads to slow wall-clock time. As depicted in Fig. 2, we profiled BERT-Base/Large models on a single A100 or A5000 GPU to illustrate the percentage of time allocated to the softmax as token lengths increase. We observe that the runtime percentage for the softmax within the entire model continues to increase *quadratically* as the token lengths grow, occupying *up to 50%* of the total model latency when the token length reaches 10^4.

Linear Attentions (LAs). Kernel-based LAs (Katharopoulos et al., 2020; Wang et al., 2020; You et al., 2023) have emerged as an effective method for eliminating the need for softmax and reducing the quadratic complexity. The core idea is to decompose the similarity measurement function, which is typically based on softmax, into separate kernel embeddings, i.e., $\mathrm{Sim}(\mathbf{Q}, \mathbf{K}) \approx \phi(\mathbf{Q})\phi(\mathbf{K})^T$. This enables us to rearrange the computation order to $\phi(\mathbf{Q})(\phi(\mathbf{K})^T\mathbf{V})$ based on the associative property of matrix multiplication. Consequently, the attention complexity is quadratic to the feature dimension d instead of the token length n. These LAs could also lead to a significant accuracy drop as compared to softmax-based attention unless carefully designed.

Autoregressive LLMs. As shown in Fig. 3, unlike the initial summarization phase, which requires processing a large number of tokens simultaneously and is thus computationally intensive, the generation phase faces severe memory or bandwidth limitations due to its autoregressive nature, involving token-by-token generation. Linear attention enables fast training and reduces the computational complexity of the summarization phase. However, it is less effective for

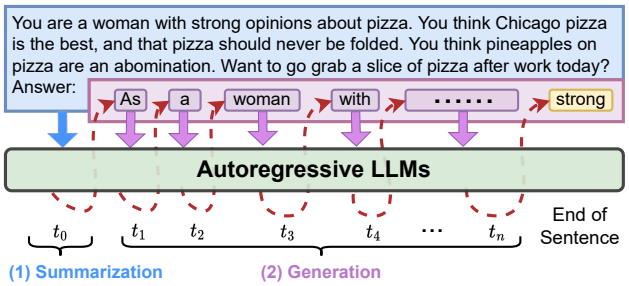

Figure 3: Illustrating the autoregressive LLMs. The process of generating text unfolds in two stages: (1) an initial *summarization phase* that employs a large batch size and utilizes the given input context, followed by (2) the *generation phase*, which operates on a single-batch basis, using previously generated tokens to continue the text output.

Table 1: Evaluation of seven LAs on BERT (Devlin et al., 2018), an encoder-based LLM, with the text classification accuracy on the GLUE benchmark (Wang et al., 2018).

BERT w/ LAs	SST2	WNLI	QNLI	MNLI	RTE	MRPC	QQP	Average
BERT (Baseline)	93.58	42.25	91.49	84.81	66.43	83.09	91.10	**78.96**
FLASH-Local	91.63	47.89	88.38	81.06	50.18	70.10	90.56	**74.26**
FLASH-Global	76.72	54.93	53.69	33.46	48.74	68.63	78.32	**59.21**
Linformer	81.54	56.34	63.06	67.54	45.13	68.38	81.32	**66.19**
Performer	80.16	45.07	60.77	39.81	45.49	67.40	75.88	**59.23**
TransNormer	81.88	56.34	67.67	67.01	53.07	70.10	83.13	**68.46**
YOSO	91.51	52.11	87.75	82.16	58.12	75.98	90.40	**76.86**
ReLU	81.77	56.34	61.54	70.14	47.29	69.85	82.44	**67.05**

autoregressive generation due to low parallelism during serving like ChatGPT (OpenAI, 2023a). Speculative decoding is one of the most critical methods for increasing parallelism, and it is imperative to establish compatibility between them to achieve both fast summarization and generation.

3.2. Evaluation of Existing LAs on LLMs

Comprehensive Evaluation. To investigate whether previous LAs can be generally applicable to three categories of LLMs: encoder-based, decoder-based, and encoder-decoder, we evaluate seven distinct LAs, including FLASH-Local&Gloabl (Hua et al., 2022), Linformer (Wang et al., 2020), Performer (Choromanski et al., 2020), TransNormer (Qin et al., 2022), YOSO (Zeng et al., 2021), ReLU (Cai et al., 2023a), across three representative LLMs in each category: encoder-based BERT (Devlin et al., 2018), decoder-based GPT-2 (Radford et al., 2019), and encoder-decoder T5 (Raffel et al., 2020). As detailed in Tabs. 1, 2, and 3, we have applied these LAs to their respective LLMs, assessing their performance across seven linguistic tasks from the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018). To enhance comparison efficacy, we also report the accuracy of softmax-based LLMs as a baseline. This facilitates a straightforward evaluation of the average accuracy drop across the seven LAs and the seven tasks when being applied to different types of LLMs.

Table 2: Evaluation of seven LAs on GPT-2 (Radford et al., 2019), a decoder-based LLM, with the text classification accuracy on the GLUE benchmark (Wang et al., 2018).

GPT-2 w/ LAs	SST2	WNLI	QNLI	MNLI	RTE	MRPC	QQP	Average
GPT-2 (Baseline)	91.28	57.75	88.39	81.54	60.65	74.51	89.13	**77.61**
FLASH-Local	83.60	53.52	77.16	73.97	48.01	68.87	86.40	**70.22**
FLASH-Global	50.92	50.70	54.27	34.59	52.35	68.38	63.19	**53.49**
Linformer	79.47	52.11	60.96	34.56	52.35	68.38	76.30	**60.59**
Performer	86.93	38.03	69.36	70.60	49.46	69.12	76.30	**65.69**
TransNormer	82.11	56.34	63.48	59.11	53.07	68.38	75.79	**65.47**
YOSO	88.42	45.07	82.23	77.80	54.51	73.04	87.72	**72.68**
ReLU	86.47	45.07	80.96	78.02	51.99	69.61	83.42	**70.79**

Table 3: Evaluation of seven LAs on T5 (Raffel et al., 2020), an encoder-decoder-based LLM, with the text classification accuracy on the GLUE benchmark (Wang et al., 2018).

T5 w/ LAs	SST2	WNLI	QNLI	MNLI	RTE	MRPC	QQP	Average
T5 (Baseline)	93.81	36.62	91.73	86.54	58.12	80.64	90.89	**76.91**
FLASH-Local	77.87	56.34	58.87	49.44	52.71	68.38	75.62	**62.75**
FLASH-Global	80.62	56.34	63.65	49.87	46.93	68.38	79.29	**63.58**
Linformer	51.15	43.66	55.43	46.60	51.99	68.38	74.19	**55.91**
Performer	82.57	56.34	63.70	61.75	52.35	69.85	78.60	**66.45**
TransNormer	79.36	43.66	59.78	48.75	58.48	70.59	75.37	**62.29**
YOSO	78.33	56.34	59.55	48.64	47.65	68.38	70.87	**61.39**
ReLU	85.79	53.52	71.57	73.52	48.01	70.34	83.89	**69.52**

Result Analysis. From our comprehensive evaluation, we have observed that: (1) most LAs are effective in encoder-based LLMs, aligning with their initial design intent. However, their performance diminishes when applied to decoder-based or encoder-decoder-based LLMs. On average, seven LAs applied to encoder-based LLMs result in an average accuracy of 67.32, whereas for decoder-based or encoder-decoder-based models, the accuracy drops to 65.56 and 63.13, respectively; (2) as visualized in the left of Fig. 1, advanced LA techniques yield notable results in encoder-based LLMs but struggle to replicate these results in decoder or encoder-decoder-based LLMs. For instance, FLASH-Local (Hua et al., 2022) and YOSO (Zeng et al., 2021) register commendable scores on BERT (74.26/76.86), only 4.7/2.1 points below the original BERT baseline. However, the average accuracy significantly reduces to 70.22/72.68 for GPT-2 and further to 62.75/61.39 for T5, marking a substantial decrease of 7.39/4.93 and 14.16/15.52 points, respectively, compared to their original softmax-based attention counterparts; (3) as shown in the right of Fig. 1, LAs that are less effective in encoder-based LLMs consistently exhibit degraded performance in decoder-based and encoder-decoder-based LLMs. This trend underscores the distinct suitability of LAs for different LLM architectures.

Limitations of Existing LAs. Our aforementioned evaluation indicates that most LAs experience an accuracy drop when applied to autoregressive decoder-based LLMs in generation tasks. Moreover, advanced LA augmentation techniques, such as those involving efficient depthwise convolution (DWConv) in the $\mathbf{V}$ (value) branch of attention modules (You et al., 2023; Xiong et al., 2021), actually fail in autoregressive LLMs due to an information leakage issue

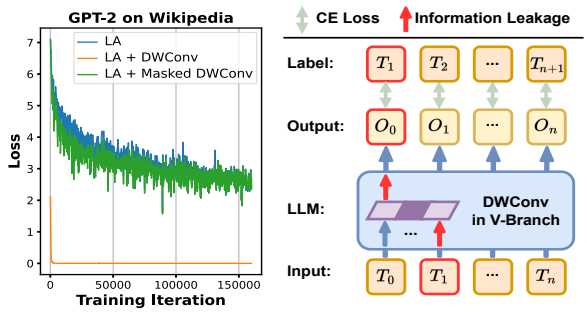

Figure 4: Existing augmented LAs fail in autoregressive LLMs. **Left**: The augmented DWConv branch results in zero loss/accuracy, as indicated by the yellow line. **Right**: Illustration of the information leakage phenomenon, i.e., next tokens are prematurely revealed as shown by red arrows, in autoregressive LLMs with DWConv in the $\mathbf{V}$ branch.

stemming from the inclusion of future context during training. As evident in Fig. 4, where LA with DWConv shows early training convergence to zero loss, yet the actual evaluation accuracy remains zero, indicating the information leakage as also illustrated in the right of Fig. 4. In addition, while LAs are beneficial in training and summarization, their effectiveness is limited in token-by-token generation. Their compatibility with speculative decoding, aimed at increasing parallelism during generation, remains a challenge. We will further discuss our augmented methods for autoregressive LLMs and their full integration with speculative decoding in the subsequent sections.

4. The Proposed Method

In this section, we introduce a revised local augmentation technique for existing LAs to enhance accuracy and examine the synergy of augmented LAs with speculative decoding for both efficient LLM training and autoregressive generation.

4.1. LA Augmentation for Autoregressive LLMs

Revised LA Augmentation. To address the information leakage problem as analyzed before, we propose to design an effective masked DWConv instead of using a simple convolutional layer for enhancing the locality of the linear attention, as used in prior studies (You et al., 2023; Xiong et al., 2021). Specifically, we adopt a causal mask on the DWConv layer to prevent tokens from accessing information from subsequent tokens, thereby preserving the inherent causality of the original attention mechanism, as illustrated in the right branch of Fig. 5. The masked DWConv prevents information leakage, contributing to better loss convergence, as demonstrated in the left of Fig. 4. Different from (Dauphin et al., 2017), our efficient DWConv is integrated directly into the attention block, rather than functioning as a standalone component. Moreover, we built our DWConv augmentation on top of existing grouped LAs to better speed up the linearized LLMs. The reason why

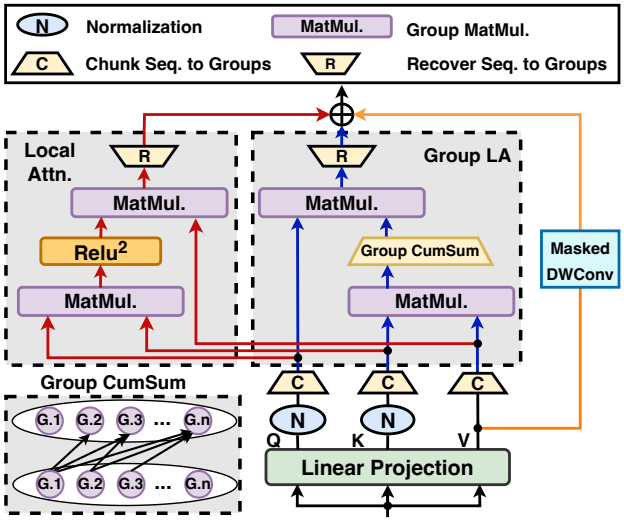

Figure 5: Model architecture of our LA augmentation.

we need the grouped LA is that standard LAs exhibit reduced efficiency in autoregressive settings due to the causal constraint (Hua et al., 2022). For example, the query vector $\mathbf{Q_t}$ at t-th time step interacts with the cumulative sum of all preceding results $\sum_{i=1}^{t} \mathbf{K_i V_i}$. This cumulative sum (`cumsum`) of $\mathbf{KV}$ product operations inherently creates a sequential dependency, and restricts the potential for parallel processing. To enhance efficiency, we follow grouped LAs to partition the input sentence into non-overlapping groups. Within each group, we bypass local dependencies, allowing parallel processing. For interactions between groups, we only compute the cumulative sums at the group level for the $\mathbf{KV}$ products for improved efficiency, as depicted in the middle branch of Fig. 5. Furthermore, to improve local dependency handling, we employ parallel local attention within each group, using softmax-based attention, as depicted in the left branch of Fig. 5. The integration of this local attention strategy with our revised local augmentation contributes significantly to the performance, combining the efficiency of LAs with improved accuracy.

Verification on Small- and Large-Scale LLMs. We evaluate and verify the revised LA augmentation on both small- and large-scale LLMs, i.e., FLASH (Hua et al., 2022) and LLaMA-7B (Touvron et al., 2023a). For FLASH, we train a small model from scratch for 100K steps on enwik8 (Hutter, 2012). As shown in Fig. 6 (a), grouped LA leads to reduced accuracy or increased loss. Local LA alone is also ineffective. A combination of grouped and local LAs showed some improvement but remained inferior to the traditional softmax-based attention method. In contrast, our augmented LAs, blending the grouped LA concept with masked DWConv augmentation (with a kernel size of 63), achieved the most favorable results among all LAs, on par with the original softmax-based attentions. For LLama-7B, We finetune it using LAs on the RedPajama dataset (Computer, 2023)

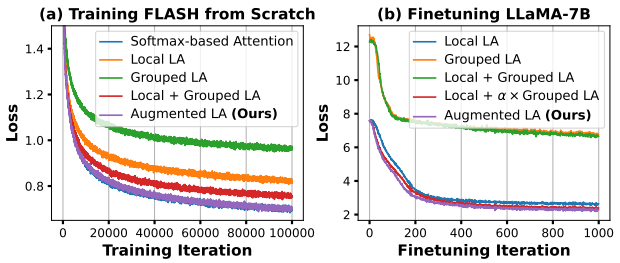

Figure 6: Tested our augmented linear attention mechanism for both training from scratch and fine-tuning from pre-trained model settings, where **(a)** shows the training progress of FLASH models (Hua et al., 2022); **(b)** depicts the fine-tuning performance of LLaMA-7B (Touvron et al., 2023a).

for 1K steps with a batch size of 64 following (Chen et al., 2023b). Fig. 6 (b) indicates a similar trend to FLASH, where local augmentation proves even more vital in this finetuning phase, and reliance solely on global LA leads to significantly higher loss. Note that we use a hyperparameter α to balance the interplay between global and local LAs. Overall, our augmented LAs combining the three branches in Fig. 5 consistently outperform existing LAs.

4.2. When LA Meets Speculative Decoding

To address the issue of limited parallelism in LLM serving, our goal is to combine speculative decoding with our enhanced LAs. Yet, a direct integration proves ineffective. Below, we explore the compatibility challenges and propose solutions that are designed to work seamlessly together.

Compatibility Analysis. Speculative decoding, such as Medusa (Cai et al., 2023b), involves using smaller draft models, e.g., multiple heads, to simultaneously predict multiple output tokens across different time steps, as illustrated in Fig. 8 (a). The original LLMs then act as verifiers, either accepting or rejecting these predictions and, if necessary, resampling them as illustrated in Fig. 8 (b). This approach enhances parallelism during LLM generation. However, combining LAs with speculative decoding presents challenges because speculative decoding generates multiple candidate outputs for each step, with varying counts per time step, altering the temporal dependency. This change is not effectively captured by masked DWConvs and grouped LAs in our augmented LAs. As shown in Fig. 7 (a), using a masked DWConv with a kernel size of 3 to convolve over stacked candidate tokens at time step t_1 results in capturing time steps $\{t_1, t_1\}$, rather than the correct sequence $\{t_0, t_1\}$. This discrepancy occurs because, at time step t_1, two candidate tokens are included in the convolution instead of the final verified one, leading to a temporal misalignment.

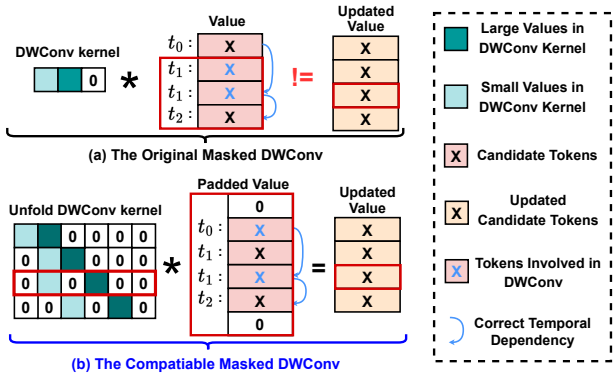

Figure 7: **(a)**: DWConv itself fails to capture the temporal dependency in speculative decoding; **(b)**: Our Unfolded DWConv kernels capture the correct temporal dependency.

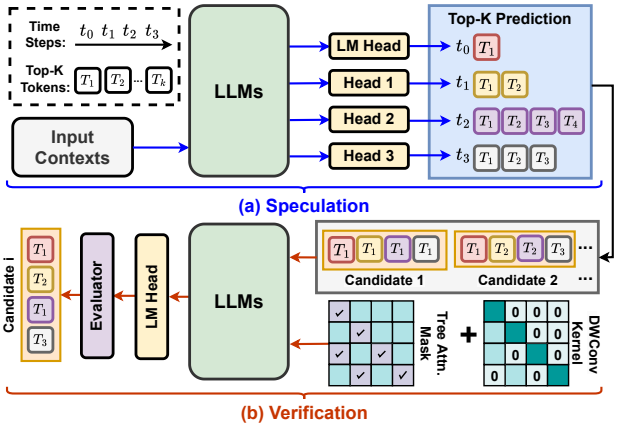

Figure 8: Illustrating the speculative decoding pipeline with our augmented LAs: (a) Speculation; and (b) Verification.

Proposed Solution. To integrate our augmented LAs with the speculative decoding, we propose the updated design of DWConv and grouped LA to take into consideration the temporal dependencies represented in Medusa's tree-based attention mask. This design ensures the simultaneous processing of multiple candidate tokens while ensuring that each token only accesses information from its preceding token. As shown in Fig. 7 (b), we unfold the convolution into matrix multiplication, akin to the img2col method (Vasudevan et al., 2017). This unfolding allows for the integration of tree-based attention masks with DWConv kernels, addressing their compatibility with negligible overheads. For example, using a masked DWConv with an unfolded kernel to convolve over stacked candidate tokens at time step t_1 successfully captures the correct sequence $\{t_0, t_1\}$, while omitting an unchosen candidate at the same time step t_1. In addition, we categorize speculative tokens into groups based on temporal dependency, regardless of the number of candidates per time step. In this way, tokens in each group interact only with verified tokens from previous groups, aligning their visibility with the tree-based attention pattern.

5. Experiments

5.1. Experiment Settings

Models, Tasks, and Datasets. *Models.* We apply our proposed augmented LA on top of five models, includ-

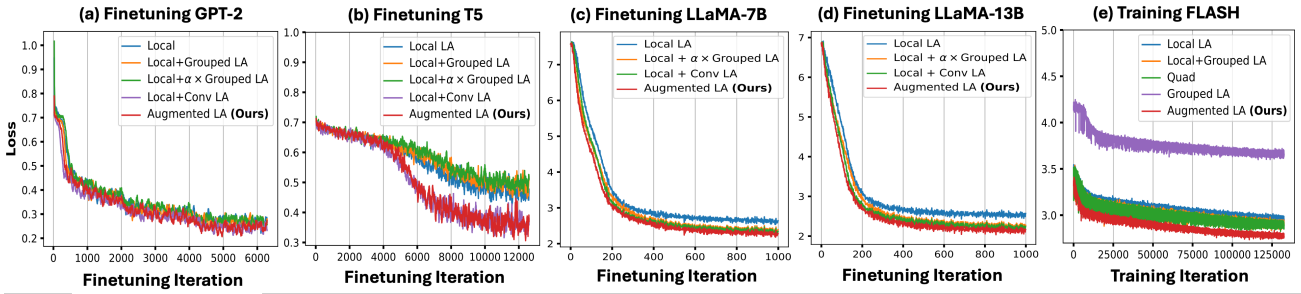

Figure 9: Visualizing the training trajectories of baseline LAs and our augmented LAs.

ing FLASH (Hua et al., 2022), T5 (Raffel et al., 2020), GPT-2 (Radford et al., 2019), LLaMA-2-7B, and LLaMA-2-13B (Touvron et al., 2023b). In particular, we train the FLASH (Hua et al., 2022) model of roughly 110M parameters from scratch and finetune the remaining language models of different sizes with our augmented LAs. *Tasks and Datasets.* For FLASH and LLaMA-2-7B/13B models, we evaluate them on language modeling tasks. Specifically, we train the FLASH model on the English partition of Wiki40b (Guo et al., 2020), which includes about 40B characters from 19.5M pages obtained from Wikipedia. We finetune the LLaMA-2-7B/13B models on RedPajama (Computer, 2023) dataset with about 1.2T tokens for 1K steps, following the setting of LongLora (Chen et al., 2023b). For T5 and GPT-2 models, we consider the text classification task to evaluate our augmented LAs, and choose seven datasets from GLUE (Wang et al., 2018) benchmark: SST2 (Socher et al., 2013), WNLI (Levesque et al., 2012), QNLI (Rajpurkar et al., 2016), MNLI (Williams et al., 2018), RTE (Dagan et al., 2006), MRPC (Dolan & Brockett, 2005), and QQP (Chen et al., 2017).

Training Settings. *For the FLASH training task*, we train the model of roughly 110M parameters from scratch with a sequence length of 1024. The batch size is 256 and the token per batch is set to 2^{18}. We use the AdamW optimizer with linear learning rate decay and a peak learning rate of 7×10^{-4}, the momentum of the AdamW optimizer is set to $\beta_1 = 0.9$, $\beta_2 = 0.95$ and the group size is set to 256 following (Hua et al., 2022). *For the LLaMA-2 finetune task*, we train it for 1K steps with a peak learning rate of 2×10^{-5} and a batch size of 64. The learning rate scheduler is constant with 20 warmup steps. The optimizer is AdamW with the momentum of $\beta_1 = 0.9$ and $\beta_2 = 0.95$. The group size is set to 64 following (Chen et al., 2023b). *For the GLUE task*, We finetune the models for 3 epochs with a learning rate of 2×10^{-5} and a batch size of 32 (Devlin et al., 2018). The group size is set to 64, and the sequence length is set to 256.

Baseline and Evaluation Metrics. *Baselines*. For the text classification task on the GLUE benchmark, we compare the proposed augmented LAs with FLASH-Local&Global (Hua et al., 2022), Linformer (Wang et al., 2020), Performer (Choromanski et al., 2020), TransNormer (Qin et al., 2022), YOSO (Zeng et al., 2021), ReLU (Cai et al., 2023a). For the LLaMA-2 finetune tasks, we compare the proposed augmented LAs with the local and global attention proposed in (Hua et al., 2022), i.e., FLASH-Local/Global. For the FLASH training task, we compare our proposed method with local, global, and quadratic softmax-based attention. *Evaluation Metrics*. For the GLUE task, we use the classification accuracy to evaluate the augmented LA and baselines. For the LLaMA-2 finetune task, we use the perplexity on PG-19 (Rae et al., 2019) to evaluate all methods. For the FLASH training task, we use the validation set perplexity of Wiki40B to evaluate. In addition, to evaluate the speedups after integrating our LAs and speculative decoding, we test the decoding speeds on MT-Bench (Zheng et al., 2023) following (Cai et al., 2023b).

Table 4: Evaluation of augmented LAs on T5 and GPT-2, with the classification accuracy on the GLUE benchmark.

GPT-2 w/	SST2	WNLI	QNLI	MNLI	RTE	MRPC	QQP	Average
LA Baseline	83.60	53.52	77.16	73.97	48.01	68.87	86.40	**70.22**
Loc.+Gro.	82.34	46.48	79.11	75.09	50.20	68.38	86.16	69.68
Loc.+α*Gro.	83.72	54.04	79.15	73.76	46.68	69.61	86.11	70.44
Augmented LA	84.72	54.93	80.01	74.26	50.90	69.85	86.16	**71.55**
T5 w/	**SST2**	**WNLI**	**QNLI**	**MNLI**	**RTE**	**MRPC**	**QQP**	**Average**
LA Baseline	77.87	56.34	58.87	49.44	52.71	68.38	75.62	**62.75**
Loc.+Gro.	76.95	56.34	60.37	51.20	49.82	68.38	75.44	62.64
Loc.+α*Gro.	78.10	56.34	59.62	51.49	49.10	68.38	75.62	62.66
Augmented LA	82.00	56.34	59.78	54.26	54.15	68.38	76.68	**64.51**

5.2. Our Augmented LAs over Baselines

Overall Comparison. We apply our augmented LAs to five decoder-based or encoder-decoder-based LLMs and compare them with other LA baselines. The training trajectories are visualized in Fig. 9. We see that our augmented LAs consistently achieve a better convergence loss as compared to all baselines. As for the quantitative results:

1. *Text Classification with GPT-2 and T5.* We evaluate the performance of the GPT-2 and T5 with our augmented LAs on the GLUE benchmark. As shown in Tab. 4, our augmented LAs consistently lead to better accuracy, e.g., on average $\uparrow$1.87 accuracy boost as compared to the competitive existing LA baselines, FLASH-Local/Global.
2. *Language Modeling with FLASH and LLaMA-7B/13B.* We evaluate the perplexity of LLaMA-7B/13B with our augmented LAs on PG-19. As shown in Tab. 5, integrating our local augmentation, i.e., masked DWConv,

Table 5: Perplexity evaluation (lower is better) on two tasks: (1) LLaMA models on PG-19 (sequence length: 4096) and (2) FLASH model on Wiki40B (sequence length: 1024).

Model	Loc.	Loc.+Gro.	Loc.+**Conv**		**Augmented LA**
LLaMA-2-7B	21.61	15.04	14.94		**13.47**
LLaMA-2-13B	19.25	12.92	12.92		**11.55**
Model	Loc.	Loc.+Gro.	Gro.	Quad.	**Augmented LA**
FLASH-110M	16.65	16.14	35.25	15.40	**15.16**

Table 6: Throughput of LLaMA (tokens/s) with LAs and the speculative decoding on MT-Bench (Zheng et al., 2023).

LLaMA w/	Loc.	Loc.+Gro.	Loc.+Conv	Loc.+Gro.+Conv
7B	32.3 (**1.0x**)	26.8 (**1.0x**)	30.4 (**1.0x**)	25.9 (**1.0x**)
7B w/ Spec.	63.3 (**2.0x**)	50.5 (**1.9x**)	55.1 (**1.8x**)	50.7 (**2.0x**)
13B	26.1 (**1.0x**)	22.7 (**1.0x**)	22.3 (**1.0x**)	20.4 (**1.0x**)
13B w/Spec.	54.4 (**2.1x**)	42.6 (**1.9x**)	47.0 (**2.1x**)	41.7 (**2.0x**)

with the local LAs results in a 6.67/6.33 reduction in perplexity. Our augmented LAs with both the local augmentation and grouped LAs, yield the lowest perplexity. The effectiveness of our augmented LAs is consistently validated by results on FLASH models and the Wiki40B dataset, demonstrating perplexity reductions ranging from 1.49 to 20.09 as compared to baselines, and even a 0.24 reduction over the original attention.

Generation Speedups by Integrating LAs with Speculative Decoding. We benchmark the speedups of our compatible LAs with speculative decoding. As shown in Tab. 6, we test the LLaMA-7B/13B models which are adapted into a chat model format, similar to LongLora (Chen et al., 2023b). Following Medusa (Cai et al., 2023b), we train Medusa heads for speculative decoding. Speed tests for the 7B and 13B models are conducted on a single A100-80GB GPU, we observe that our revised LAs are compatible with speculative decoding and approximately doubled the speed.

5.3. Ablation Study

Our LA Speedups. We benchmark the training speed of FLASH with original attention or our augmented LAs. Specifically, we set the batch size as 1 and measure the training step time in seconds on a single A100-40GB GPU. As demonstrated in Tab. 7, we observe that our augmented LAs achieve faster training, e.g., 1.52×/2.94× faster than the original attention counterpart with a quadratic cost for inputs of 4K/8K sequence lengths. Note that for fair comparison, we always keep the group size in FLASH at 256 in all sequence lengths.

Table 7: Step time comparison (in seconds) between FLASH with original attention and our augmented LAs.

FLASH w/	Seq=4K	Seq=8K
Ori. Attn	1.60	5.74
Aug. LA	**1.05**	**1.95**

Breakdown Analysis of Augmented LAs. To gain insights into the contribution of each component in our augmented LAs, we show the breakdown analysis using GPT-2 and T5 models on Wikitext2 (Merity et al., 2017)/PTB (Marcus et al., 1993) and CNN/Daily Mail (See et al., 2017) datasets, respectively. As shown in Tabs. 8 and 9, our local augmentation, i.e., masked DWConv, consistently augments the local or grouped LAs, leading to 5.71 perplexity reductions on GPT-2 and 3.59 Rouge1 score (Lin, 2004) improvement on T5. Our augmented LAs, consisting of both local augmentation and grouped LAs, achieve the best results, i.e., 11.83∼17.54 perplexity reduction and 4.23∼15.45 Rouge1 score improvement, over all other LA variants.

Table 8: Perplexity of GPT-2 with our augmented LAs on the Wikitext2 and PTB datasets.

GPT-2 w/	Loc.	Loc.+Gro.	Loc.+**Conv**	**Augmented LA**
Wikitext2	56.80	42.81	51.09	39.26
PTB	69.32	57.72	84.24	46.85

Table 9: Ablation studies of fine-tuning T5 with LAs on the CNN/Daily Mail dataset (See et al., 2017).

T5 w/	**Rouge1**	**Rouge2**	**RougeL**	**RougeLsum**
Local LA	8.65	0.17	7.14	8.27
Grouped LA	6.14	0.86	5.77	5.50
Local + Grouped LA	19.87	3.07	14.54	18.29
Local + α×Grouped LA	19.01	2.90	13.99	17.54
Local LA + DWConv	12.24	0.20	8.95	11.38
Augmented LAs	24.10	4.93	17.22	22.11

Extend to Longer Sequence. We finetuned the LLaMA-2-7B model to increase its sequence length from 4096 to 8192, using our augmented LAs following the setting of LongLora (Chen et al., 2023b) on the RedPajama dataset. For a fair comparison, we only use the local attention in LongLora without shifting the attention mask with the block size of 256. As shown in Tab. 10, our proposed augmented LA helps reduce the perplexity by 1.43, showing its efficacy when extending to a longer sequence.

Table 10: Perplexity of LLaMA-2-7B under 8K sequence length.

LongLora	**Perplexity**
w/o Aug. LA	15.29
w/ Aug. LA	**13.86**

6. Conclusion

In this paper, we present the first empirical analysis of linearized LLMs, revealing the inefficiency or failure of many existing linear attention techniques in autoregressive decoding with masked attention. Moreover, we revise the local augmentation of linear attention for decoder-based autoregressive LLMs, enhancing performance and preventing information leakage. In addition, we explore how linear attention can be integrated with speculative decoding to improve the parallelism of linearized LLMs during serving or generation. Extensive experiments and ablation studies across seven linear attention methods and five encoder/decoder-based LLMs consistently validate the effectiveness of our augmented linear attentions and their seamless compatibility with speculative decoding.

Broader Impact

Efficient LLM Training and Serving Goal. The recent advancements in Large Language Models (LLMs), exemplified by OpenAI's GPT-3 with its 175 billion parameters, have underscored the significant data and computational power required for such technologies. Training models of this scale incur substantial costs, both financially and environmentally. For instance, the cost necessary to train GPT-3 could exceed 4 million equivalent GPU hours (Brown et al., 2020), and the carbon footprint of training a single Transformer model might rival the lifetime emissions of five average American cars (Strubell et al., 2019). Addressing the challenges of efficient training and serving of LLMs is therefore not only a technical imperative but also an environmental and ethical necessity.

Societal Consequences. The success of this project in enabling more efficient training and serving of LLMs will have far-reaching implications, especially in processing long sequences commonly encountered in document handling. Our efforts are set to substantially influence various societal and economic sectors. The enhanced efficiency of LLMs promises transformative changes in diverse applications ranging from document summarization and question answering to personal digital assistants, security, and augmented reality. The development and exploration of linearized LLMs mark a pivotal progress in rendering these models both more accessible and environmentally sustainable.

References

Arar, M., Shamir, A., and Bermano, A. H. Learned queries for efficient local attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10841–10852, 2022.

Bae, S., Ko, J., Song, H., and Yun, S.-Y. Fast and robust early-exiting framework for autoregressive language models with synchronized parallel decoding. arXiv preprint arXiv:2310.05424, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. Advances in neural information processing systems, 33: 1877–1901, 2020.

Cai, H., Li, J., Hu, M., Gan, C., and Han, S. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17302–17313, 2023a.

Cai, T., Li, Y., Geng, Z., Peng, H., and Dao, T. Medusa: Simple framework for accelerating llm generation with multiple decoding heads, 2023b.

Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., and Jumper, J. Accelerating large language model decoding with speculative sampling. arXiv preprint arXiv:2302.01318, 2023a.

Chen, Y., Qian, S., Tang, H., Lai, X., Liu, Z., Han, S., and Jia, J. Longlora: Efficient fine-tuning of long-context large language models. arXiv preprint arXiv:2309.12307, 2023b.

Chen, Z., Zhang, H., Zhang, X., and Zhao, L. Quora question pairs. 2017.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al. Rethinking attention with performers. arXiv preprint arXiv:2009.14794, 2020.

Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., Belanger, D. B., Colwell, L. J., and Weller, A. Rethinking attention with performers. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Computer, T. Redpajama: an open dataset for training large language models, 2023. URL https://github.com/togethercomputer/RedPajama-Data.

Dagan, I., Glickman, O., and Magnini, B. The pascal recognising textual entailment challenge. In Quiñonero-Candela, J., Dagan, I., Magnini, B., and d'Alché Buc, F. (eds.), Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment, pp. 177–190, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33428-6.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in Neural Information Processing Systems, 35:16344–16359, 2022.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. Language modeling with gated convolutional networks. In International conference on machine learning, pp. 933–941. PMLR, 2017.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

Dolan, W. B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005), 2005. URL https://aclanthology.org/I05-5002.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2021.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. Optq: Accurate quantization for generative pre-trained transformers. In The Eleventh International Conference on Learning Representations, 2022.

Guo, M., Dai, Z., Vrandečić, D., and Al-Rfou, R. Wiki-40B: Multilingual language model dataset. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S. (eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 2440–2452, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.297.

Hua, W., Dai, Z., Liu, H., and Le, Q. Transformer quality in linear time. In International Conference on Machine Learning, pp. 9099–9117. PMLR, 2022.

Hutter, M. The human knowledge compression contest. URL http://prize. hutter1. net, 6, 2012.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In International Conference on Machine Learning, pp. 5156–5165. PMLR, 2020.

Kim, S., Mangalam, K., Malik, J., Mahoney, M. W., Gholami, A., and Keutzer, K. Big little transformer decoder. arXiv preprint arXiv:2302.07863, 2023.

Levesque, H., Davis, E., and Morgenstern, L. The winograd schema challenge. In 13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012, Proceedings of the International Conference on Knowledge Representation and Reasoning, pp. 552–561. Institute of Electrical and Electronics Engineers Inc., 2012. ISBN 9781577355601. 13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012 ; Conference date: 10-06-2012 Through 14-06-2012.

Leviathan, Y., Kalman, M., and Matias, Y. Fast inference from transformers via speculative decoding. In International Conference on Machine Learning, pp. 19274–19286. PMLR, 2023.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022, 2021.

Lu, J., Yao, J., Zhang, J., Zhu, X., Xu, H., Gao, W., Xu, C., Xiang, T., and Zhang, L. Soft: softmax-free transformer with linear complexity. Advances in Neural Information Processing Systems, 34:21297–21309, 2021.

Ma, X., Fang, G., and Wang, X. Llm-pruner: On the structural pruning of large language models. arXiv preprint arXiv:2305.11627, 2023.

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. Building a large annotated corpus of english: The penn treebank. 1993.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In International Conference on Learning Representations, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Miao, X., Oliaro, G., Zhang, Z., Cheng, X., Wang, Z., Wong, R. Y. Y., Chen, Z., Arfeen, D., Abhyankar, R., and Jia, Z. Specinfer: Accelerating generative llm serving with speculative inference and token tree verification. arXiv preprint arXiv:2305.09781, 2023.

OpenAI. Chatgpt: Language model for dialogue generation. 2023a. URL https://www.openai.com/chatgpt/.

OpenAI. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023b.

Qin, Z., Han, X., Sun, W., Li, D., Kong, L., Barnes, N., and Zhong, Y. The devil in linear transformer. arXiv preprint arXiv:2210.10340, 2022.

Qin, Z., Li, D., Sun, W., Sun, W., Shen, X., Han, X., Wei, Y., Lv, B., Yuan, F., Luo, X., et al. Scaling transnormer to 175 billion parameters. arXiv preprint arXiv:2307.14995, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

Rae, J. W., Potapenko, A., Jayakumar, S. M., and Lillicrap, T. P. Compressive transformers for long-range sequence modelling. arXiv preprint arXiv:1911.05507, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ questions for machine comprehension of text. In Su, J., Duh, K., and Carreras, X. (eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In International Conference on Machine Learning, pp. 8821–8831. PMLR, 2021.

Roberts, A., Chung, H. W., Levskaya, A., Mishra, G., Bradbury, J., Andor, D., Narang, S., Lester, B., Gaffney, C., Mohiuddin, A., Hawthorne, C., Lewkowycz, A., Salcianu, A., van Zee, M., Austin, J., Goodman, S., Soares, L. B., Hu, H., Tsvyashchenko, S., Chowdhery, A., Bastings, J., Bulian, J., Garcia, X., Ni, J., Chen, A., Kenealy, K., Clark, J. H., Lee, S., Garrette, D., Lee-Thorp, J., Raffel, C., Shazeer, N., Ritter, M., Bosma, M., Passos, A., Maitin-Shepard, J., Fiedel, N., Omernick, M., Saeta, B., Sepassi, R., Spiridonov, A., Newlan, J., and Gesmundo, A. Scaling up models and data with `t5x` and `seqio`. arXiv preprint arXiv:2203.17189, 2022. URL https://arxiv.org/abs/2203.17189.

Schuster, T., Fisch, A., Gupta, J., Dehghani, M., Bahri, D., Tran, V., Tay, Y., and Metzler, D. Confident adaptive language modeling. Advances in Neural Information Processing Systems, 35:17456–17472, 2022.

See, A., Liu, P. J., and Manning, C. D. Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1073–1083, 2017.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Yarowsky, D., Baldwin, T., Korhonen, A., Livescu, K., and Bethard, S. (eds.), Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in nlp. arXiv preprint arXiv:1906.02243, 2019.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023b.

Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., and Li, Y. Maxvit: Multi-axis vision transformer. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV, pp. 459–479. Springer, 2022.

Vasudevan, A., Anderson, A., and Gregg, D. Parallel multi channel convolution using general matrix multiplication. In 2017 IEEE 28th international conference on application-specific systems, architectures and processors (ASAP), pp. 19–24. IEEE, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

Waisberg, E., Ong, J., Masalkhi, M., Zaman, N., Sarker, P., Lee, A. G., and Tavakkoli, A. Google's ai chatbot "bard": a side-by-side comparison with chatgpt and its utilization in ophthalmology. Eye, pp. 1–4, 2023.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461, 2018.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity, 2020.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In Walker, M., Ji, H., and Stent, A. (eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. Smoothquant: Accurate and efficient post-training quantization for large language models. In International Conference on Machine Learning, pp. 38087–38099. PMLR, 2023.

Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y., and Singh, V. Nyströmformer: A nyström-based algorithm for approximating self-attention. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pp. 14138–14148, 2021.

You, H., Xiong, Y., Dai, X., Wu, B., Zhang, P., Fan, H., Vajda, P., and Lin, Y. C. Castling-vit: Compressing self-attention via switching towards linear-angular attention at vision transformer inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14431–14442, 2023.

Zeng, Z., Xiong, Y., Ravi, S., Acharya, S., Fung, G. M., and Singh, V. You only sample (almost) once: Linear cost self-attention via bernoulli sampling. In International conference on machine learning, pp. 12321–12332. PMLR, 2021.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. arXiv preprint arXiv:2306.05685, 2023.

Zhu, C., Ping, W., Xiao, C., Shoeybi, M., Goldstein, T., Anandkumar, A., and Catanzaro, B. Long-short transformer: Efficient transformers for language and vision. Advances in Neural Information Processing Systems, 34, 2021.

A. More Visualization of Training Trajectories.

As detailed in Sec. 5.3, we present a quantitative analysis comparing local LAs, grouped LAs, and our augmented LAs that combine both local augmentation and grouped LAs. This appendix provides the training trajectories for GPT-2 using these LA methods. Fig. 10 demonstrates that our local augmentation, specifically masked DWConv, effectively enhances both local and grouped LAs. Moreover, our augmented LAs, which integrate local augmentation with grouped LAs, exhibit the most favorable convergence in terms of loss.

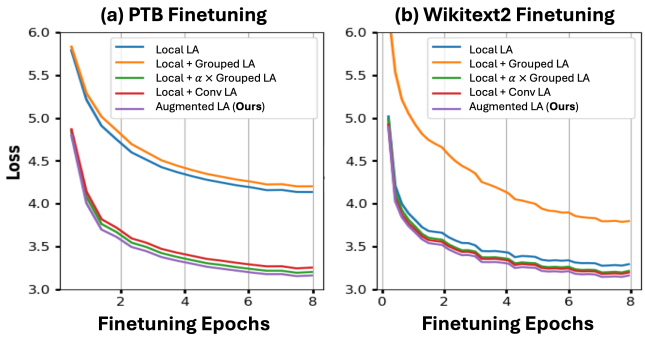

Figure 10: Visualizing the training trajectories of baseline LAs and our augmented LAs.