TRANSFORMER INSTABILITY IN LONG SEQUENCE TRAINING: THE UNDERESTIMATED ROLE OF SHORT-RANGE DEPENDENCIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformer language models have driven remarkable progress across diverse fields, including natural language processing, speech processing, and computer vision. However, despite extensive research, transformers remain prone to training instability on long sequences, often manifesting as sudden spikes or divergence in the training loss during a run.

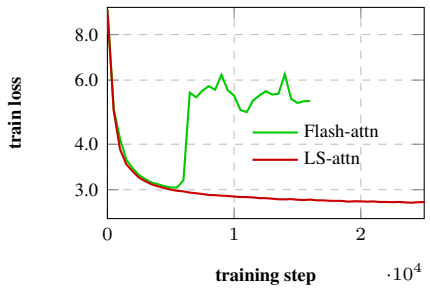
In this work, we identify a key source of this instability: self-attention's limited capacity to capture short-range dependencies – particularly in tasks such as language modeling, where most tokens depend heavily on their immediate neighbors. This limitation leads to rapid growth of the self-attention's logits during long-sequence training, ultimately destabilizing optimization.

To address this, we propose augmenting the standard architecture with several local (short-range) attention heads alongside the full (long-range) attention heads. The local heads explicitly capture short-range dependencies, while the full heads preserve long-range context. This composed self-attention – termed Long Short-attention (LS-attention) – stabilizes training by mitigating logit explosion. Across a wide range of experiments, we demonstrate that long-sequence training triggers logit explosion for multi-head self-attention (MHSA), whereas LS-attention effectively prevents it. Additionally, LS-attention makes transformer models more efficient, reducing inference latency by up to 44% compared to equivalent state-of-the-art MHSA implementations.

1 INTRODUCTION

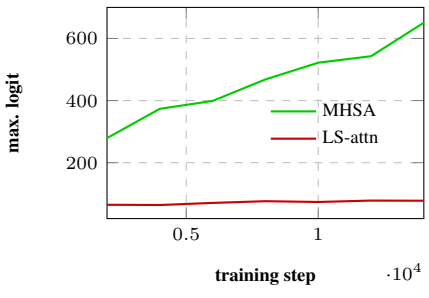
Transformer language models have become the backbone of modern machine learning systems, achieving remarkable success across diverse domains such as natural language processing (Vaswani et al. (2017); Devlin et al. (2019); Radford et al. (2018; 2019)), computer vision (Chen et al. (2020); Yu et al. (2022); Pippi et al. (2025); Chang et al. (2022)), and speech (Baevski et al. (2020); Hsu et al. (2021); Ao et al. (2022); Gulati et al. (2020)). These models have enabled state-of-the-art results in applications like machine translation, document summarization, code generation, image captioning, and multimodal reasoning.

Despite their immense success, transformer language models often exhibit training instability, particularly during large-scale pretraining or when processing long sequences (Molybog et al. (2023); Chowdhery et al. (2023); Li et al. (2022); Wortsman et al. (2024); Zhai et al. (2023); Dehghani et al. (2023); Nishida et al. (2024); Wang et al. (2025); Kedia et al. (2024)). This instability typically manifests as spikes or divergence in the training loss. Several explanations and solutions for this training instability have been proposed in the literature. For instance, Liu et al. (2020) attribute instability to the amplification of small parameter perturbations due to reliance on the residual branch. Others, such as Molybog et al. (2023), implicate the Adam optimizer (Kingma & Ba (2015)) as a contributing factor. The use of long sequences during training has also been linked to instability, prompting strategies like progressive sequence length increase during training (Li et al. (2022; 2021)). Several studies, such as Wortsman et al. (2024); Zhai et al. (2023); Dehghani et al. (2023); Kedia et al. (2024), associate the issue with logit explosion and propose normalization techniques (e.g., QK-norm Henry et al. (2020)) to stabilize training, though the root cause of the explosion remains



t

(a) Plot of training loss over training steps.



(b) Plot of maximum absolute logit from attention operations over training steps.

Figure 1: Illustration of training instability and logit explosion in Flash-attention, a state-of-the-art implementation of MHSA. The left plots show that the training loss of an autoregressive transformer with Flash-attention begins to diverge after a certain number of steps for longer sequences ($n = 2K$), whereas the same model with LS-attention (incorporating both local and full attention heads) remains stable. The right plots compare the maximum absolute logits of MHSA and LS-attention during training. LS-attention mitigates logit explosion by reducing the maximum logit magnitude to less than one-eighth of that in MHSA.

unclear. Nishida et al. (2024) identify norm imbalance among parameters as a source of instability and introduce reparameterization methods to address it. Additional techniques such as learning rate warm-up, weight decay, and μParam (Yang et al. (2022)) have also been explored. However, a clear understanding of the underlying causes – particularly those stemming from the behavior of the self-attention mechanism – and their effective mitigation remains an active area of research.

Cause of Instability: Although several studies (e.g., Wortsman et al. (2024); Zhai et al. (2023); Dehghani et al. (2023); Kedia et al. (2024)) have identified the explosion of logits in self-attention as a key contributor to training instability, the underlying cause of this phenomenon remains largely unexplained. In this work, we attribute the logit explosion to self-attention's limited capacity to model local or short-range dependencies – especially in tasks such as natural language processing, where almost every token typically relies heavily on its neighboring tokens. To elaborate, let $\mathrm{X} = [\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}]^T \in \mathbb{R}^{n \times d}$ represents a sequence of n input tokens. The self-attention mechanism transforms X into new representations $\mathrm{Y} = [\mathbf{y}_0, \ldots, \mathbf{y}_{n-1}]^T \in \mathbb{R}^{n \times d}$, computed as:

$$\mathrm{Y} = \mathrm{PXW}_v,$$

where $\mathrm{W}_v \in \mathbb{R}^{d \times d}$ is a trainable weight matrix, and $\mathrm{P} \in \mathbb{R}^{n \times n}$ is the attention matrix encoding the token dependencies. Each row of P is a probability distribution, where a high $\mathrm{P}[i, j]$ implies that the representation $\mathbf{y}_i$ strongly incorporates information from $\mathbf{x}_j$. The attention matrix is computed via: $\mathrm{P} = \mathrm{softmax}(\mathrm{S}) = \mathrm{softmax}(\mathrm{QK}^T) = \mathrm{softmax}(\mathrm{XW}_Q\mathrm{W}_K^T\mathrm{X}^T)$[1], where $\mathrm{Q}, \mathrm{K} \in \mathbb{R}^{n \times d}$ are the query and key matrices, respectively, and $\mathrm{S} \in \mathbb{R}^{n \times n}$ contains the pre-softmax logits. To model arbitrary dependencies between n tokens, the attention matrix P ideally requires $O(n^2)$ parameters. However, because P is derived from the product of two $n \times d$ matrices, its number of parameters remains $O(nd)$. When $n \gg d$, this becomes a low-rank bottleneck (Bhojanapalli et al. (2020)). In tasks where all tokens depends on a small set of "keyword" tokens, the attention matrix becomes low-rank, which is well repressented by $O(nd)$ parameters. However, in tasks requiring dense local dependencies – where nearly every token depends on its immediate neighbors – the attention matrix must be effectively high-rank (as shown in Figure 2a). The difficulty of high rank attention matrix using only $O(nd)$ parameters forces the model to compensate by inflating the logits S when $d \ll n$, leading to training instability in these scenarios.

The Solution: The key idea behind our approach to mitigating logit explosion stems from the observation that local dependencies typically span only a small window around each token. As a result, they can be effectively captured using $O(nl)$ parameters, where $l \ll n$ denotes the local

[1]Without loss of generality, we ingore the logit scaling factor for simplicity.

window size. In contrast, full attention attempts to model interactions between all pairs of n tokens in the input sequence, requiring the representation of $O(n^2)$ attention weights. This demand often exceeds the expressive capacity of the attention mechanism, since its parameterization is limited to $O(nd)$. A sliding-window local attention mechanism, which restricts each query token's attention span to a small neighborhood of l' tokens ($l' \ll n$), reduces the number of attention scores to be represented to $O(nl')$, making it more compatible with the available parameterization. Local attention is therefore more effective than full attention for capturing dense short-range dependencies. However, local attention alone is insufficient for modeling long-range dependencies, which remain essential for strong performance for many tasks. To meet both needs, we propose using both local (short-range) and full (long-range) attention heads. This composed attention, referred to as LS-attention, enables transformer models to effectively capture both short- and long-range dependencies while reducing the risk of logit explosion during training (as illustrated in Figure 1).

Efficiency of The Solution: In addition to improving training stability, LS-attention offers computational efficiency during both training and inference. For longer sequences, the computational overhead of a transformer model is dominated by the MHSA module, which uses full attention heads with quadratic computational complexity in the sequence length n. In contrast, a local attention head with attention span $l \ll n$ exhibits nearly linear complexity with respect to n. In practice, we find that LS-attention, with only a few full attention heads and the remaining heads as local attention, performs very well, which reduces both training and inference time significantly. In our experiments, we found that a model using LS-attention was up to 44% more efficient during inference compared to one using Flash-attention (Dao et al. (2022); Dao (2024)), the state-of-the-art efficient implementation of MHSA, on longer sequences.

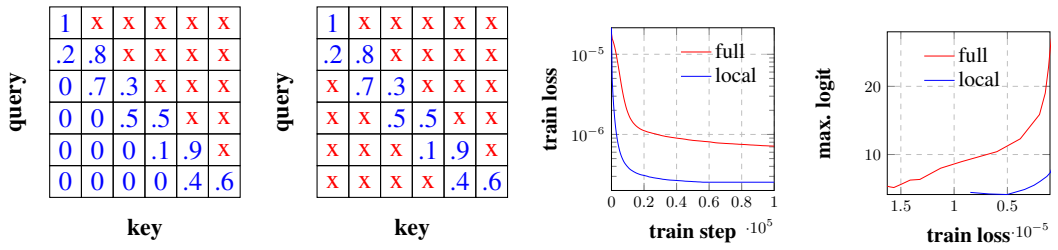
Summary of Contributions: The contributions of this work are summarized as follows:

- We identify a key limitation of self-attention: its inability to effectively model dense local dependencies in long sequences. This limitation can lead to logit explosion during long-sequence training, contributing to training instability in transformer models, particularly for tasks such as language modeling.
- To address the above limitation, we propose to compose local (short-range) and full (long-range) attention heads. The composed self-attention, referred to as Long Short-attention (LS-attention), mitigates the logit explosion and stabilizes the training. Through extensive experimentation, we validate that long sequence training leads to logit explosion in MHSA while LS-attention mitigates it.
- We have also investigated the applicability of other structured self-attention mechanisms and training stabilization methods for stabilizing long-sequence training.

2 Related Works

Low-Rank Bottleneck of Self-Attention Earler, Bhojanapalli et al. (2020) identified a low-rank bottleneck in the self-attention layer, showing that it may not represent all possible attention matrices P when the embedding dimension $d < n$. To address this, they proposed setting the head dimension to n. However, this strategy becomes impractical for large n, as it increases the computational complexity of the MHSA operation to $O(Hn^3)$, where H is the number of heads. In contrast, our work identifies a specific scenario – the presence of dense local dependencies – where this low-rank bottleneck leads to critical training instability in transformer networks. Based on this observation, we propose an efficient solution that not only resolves the instability but also improves computational efficiency on long sequences.

Structured Self-Attention Various structured self-attention mechanisms have been extensively explored to mitigate the quadratic computational complexity of vanilla self-attention. For example, prior works, such as Child et al. (2019); Beltagy et al. (2020); Zaheer et al. (2020); Jiang et al. (2024); Guo et al. (2019), have proposed replacing full self-attention with combinations of sparse self-attentions – such as local, global, and dilated attention – to overcome the quadratic complexity barrier and improve model efficiency on long sequences. In contrast, our work identifies a key limitation of full attention in capturing dense local dependencies and demonstrates that structured



(a) Attention pattern for (causally) full attention.

(b) Attention pattern for local attention.

(c) Training curve for syn. experiment.

(d) Maximum logit vs. train loss on syn. experiment.

Figure 2: Comparison of representing dense local dependencies by local and full attention. (a) Full attention attempts to represent $O(n^2)$ attention scores (shown in blue) using only $O(nd)$ degrees of freedom. (b) Local attention focuses on $O(nl')$ attention scores, where $l' \ll n$, making it a better fit for the available $O(nd)$ capacity. (c) In a synthetic dense local dependency learning task, local attention achieves lower training loss. (d) Local attention requires lower logit values than full attention to achieve the same training loss.

self-attention can be used to address this limitation. Thus, our contribution is to develop efficient self-attention that enhances – rather than compromises – the representational power of full attention.

3 Impact of Dense Local Dependencies on Logit Explosion in Long-Sequence Training

In this section, we analyze the ability of (full) self-attention to learn dense local dependencies. To this end, consider an autoregressive task over sequences of length n, where the prediction of the next token depends only on the immediately preceding l tokens, with $l \ll n$. For this task, the ideal attention matrix $\mathrm{P} \in \mathbb{R}^{n \times n}$ would satisfy $\mathrm{P}[i,j] > 0$ for $0 \leq i - j \leq l$, and $\mathrm{P}[i,j] = 0$ otherwise.

When attempting to learn this dependency pattern using full causal attention, the model aims to approximate a matrix P' using the self-attention operation such that $\mathrm{P}'[i,j] = \mathrm{P}[i,j]$ for $i - j \geq 0$, and treats $\mathrm{P}'[i,j]$ as a "don't care" term for $i - j < 0$ (since these terms are masked in causal attention). An illustration of such an attention pattern is shown in Figure 2a, where $n = 6$ and $l = 2$; red entries represent masked (don't care) terms. Importantly, P' is a matrix of rank n, which grows linearly with the sequence length. During training, the attention mechanism attempts to replicate P' using $\mathrm{softmax}((\mathrm{QK}^T + \mathrm{M}_{\mathcal{S}})/\sqrt{d})$, where $\mathrm{Q} = [\mathbf{q}_0, \ldots, \mathbf{q}_{n-1}]^T \in \mathbb{R}^{n \times d}$, $\mathrm{K} = [\mathbf{k}_0, \ldots, \mathbf{k}_{n-1}]^T \in \mathbb{R}^{n \times d}$ be the query and key matrices and $\mathrm{M}_{\mathcal{S}} \in \mathbb{R}^{n \times n}$ be the causal mask (i.e., $\mathrm{M}_{\mathcal{S}}[i,j] = 0$ for $i - j \geq 0$ and $-\infty$ otherwise). However doing so requires representing $O(n^2)$ non-masked entries in P' using only Q and K of $O(nd)$ dimension. This mismatch becomes a critical bottleneck in settings where $n \gg d$, leading to logit explosion and training instability.

A sliding window local attention does not suffer from the same limitations when capturing such local dependencies. It attempts to reconstruct the ideal attention matrix P only for the subset of entries $\{(i,j) : 0 \leq i - j \leq l'\}$, where the local attention span $l' \ll n$ and is on the same order as l. An example of an attention pattern learned by a sliding window local attention is shown in Figure 2b. In this case, the attention mechanism needs to learn only $O(nl')$ entries, which is significantly smaller than $O(n^2)$ for full attention. As a result, local attention is better suited for learning dense local dependencies compared to full attention.

3.1 Validation through a Synthetic Task

Our synthetic task is designed to evaluate the representational power of the self-attention, $\mathrm{Y} = \mathrm{softmax}((\mathrm{QK}^T + \mathrm{M}_{\mathcal{S}})/\sqrt{d})\mathrm{V}$ (where $\mathrm{V} = [\mathbf{v}_0, \ldots, \mathbf{v}_{n-1}]^T \in \mathbb{R}^{n \times d}$ be the value matrix), in capturing local dependencies when Q and K are allowed to freely take any values. The goal is to predict the output $\mathrm{Y} = [\mathbf{y}_0, \ldots, \mathbf{y}_{n-1}]^T \in \mathbb{R}^{n \times d}$ of a sequence given the input V, such that Y

satisfies $\mathrm{Y} = \mathrm{PV}$ for a ground truth attention matrix $\mathrm{P} \in \mathbb{R}^{n \times n}$. The matrix P is constructed to encode dense local dependencies, typically as a banded matrix where only entries within a fixed window l around the diagonal can be non-zero. Therefore, predicting Y from V using an attention mechanism effectively requires learning Q and K such that $\mathrm{P} \approx \mathrm{softmax}((\mathrm{QK}^T + \mathrm{M}_{\mathcal{S}})/\sqrt{d})$ is satisfied, where $\mathrm{M}_{\mathcal{S}}$ denotes the appropriate masking matrix for full and local attention.

To that end, we generated a 2500×2500 ground truth attention matrix P such that

$$P[i,j] = \begin{cases} p_{ij}, & \text{if } 0 < i - j \leq 50 \\ 0, & \text{otherwise} \end{cases}$$

where each p_{ij} is independently drawn from a Bernoulli distribution with probability 0.5. The matrix P is then row-normalized to ensure it represents a valid attention distribution. We set V to be the identity matrix of size 2500×2500, so that each $\mathbf{y}_i$ can be expressed as a unique linear combination of the $\mathbf{v}_j$s. This setup guarantees the uniqueness of P in the relation $\mathrm{Y} = \mathrm{PV}$.

We trained both full and local self-attention operations for $100K$ steps using the Adam optimizer, with the key/query dimensionality set to 25. For the local attention, we used a sliding window of span 50. The training losses for both models are shown in Figure 2c. As illustrated, local attention leads to faster convergence and achieves significantly lower training loss compared to full attention after $100K$ steps, indicating its superior ability to model dense local dependencies. Additionally, we plot the maximum logit values of both self-attention mechanisms against their corresponding training losses, as shown in Figure 2d. The figure shows that local attention attains the same training loss with significantly smaller logits, underscoring full attention's greater susceptibility to the logit explosion problem when modeling local dependencies.

4 Long Short-attention: The Solution to Logit Explosion

Since local attention mechanisms are better suited than full attention for capturing dense local dependencies and induce less logit explosion, we propose the use of dedicated local attention heads to explicitly capture short-range interactions. However, local attention alone cannot capture the long-range dependencies. To overcome this limitation, our approach integrates local and full attentions, thereby enabling the joint modeling of both short- and long-range dependencies. We rely on the assumption that the overall attention matrix P can be approximately decomposed as $\mathrm{P} \approx \mathrm{P}_{S_0} + \cdots + \mathrm{P}_{S_{H_s-1}} + \mathrm{P}_{L_0} + \cdots + \mathrm{P}_{L_{H_l-1}}$ where each P_{S_i} captures local dependencies within a small attention span $p \ll n$, and each P_{L_j} captures long-range dependencies. Each P_{L_j} is assumed to be low-rank. This assumption is motivated by the observation that, in many applications, only a small number of "keyword" tokens receive attention in long-range interactions, resulting in low-rank long-range attention patterns.

Given such a decomposition, the attention output can be approximated as:

$$\begin{aligned} \mathrm{Y} = \mathrm{PV} &\approx \sum_{i=0}^{H_s-1} \mathrm{P}_{S_i}\mathrm{V} + \sum_{i=0}^{H_l-1} \mathrm{P}_{L_i}\mathrm{V} \\ &\approx \sum_{i=0}^{H_s-1} \mathrm{softmax}\left(\left(\mathrm{Q}_{S_i}\mathrm{K}_{S_i}^T + \mathrm{M}_s\right)/\sqrt{d}\right)\mathrm{V} + \sum_{i=0}^{H_l-1} \mathrm{softmax}\left(\left(\mathrm{Q}_{L_i}\mathrm{K}_{L_i}^T + \mathrm{M}_l\right)/\sqrt{d}\right)\mathrm{V} \end{aligned}$$

where M_s and M_l are the attention masks for short-range and long-range attention, respectively. In practice, we implement this combined mechanism using a $(s+l)$-head attention module, referred to as Long Short-attention (LS-attention), with s short-range (local) attention heads and l long-range (full) attention heads. Therefore, the output of LS-attention is given by:

$$\begin{aligned} \text{LS-attn}(\mathrm{X}) &= \mathrm{Concat}(\mathrm{O}^{(0)}, \ldots, \mathrm{O}^{(H-1)})\mathrm{W}_O, \text{ such that} \\ \mathrm{O}^{(i)} = \mathrm{softmax}\left((\mathrm{Q}^{(i)}{\mathrm{K}^{(i)}}^T + \mathrm{M}^{(i)})/\sqrt{d}\right)\mathrm{V}^{(i)} &= \mathrm{softmax}\left((\mathrm{X}\mathrm{W}_Q^{(i)}{\mathrm{W}_{\mathrm{K}}^{(i)}}^T\mathrm{X}^T + \mathrm{M}^{(i)})/\sqrt{d}\right)\mathrm{X}\mathrm{W}_V^{(i)} \end{aligned}$$

where $H = s + l$, $\mathrm{O}^{(i)}$ is the output of the i-th attention head, $\mathrm{M}^{(i)}$ is the attention mask matrix for the i-th attention, and set to local attention mask for the first s heads and to the full attention

mask (such as causal attention) for the last l heads. In practice, we do not implement the LS-attention using the above parallel form. Rather, we use the efficient self-attention implementation of Dao et al. (2022); Dao (2024); Shah et al. (2024).

RUNTIME AND MEMORY REQUIREMENTS

A full attention head requires $O(n^2d)$ FLOPs. In contrast, a local attention head with an attention span of p requires only $O(npd)$ FLOPs. Therefore, an LS-attention module with s local heads and l full heads requires approximately $O(n(sp + nl)d) \approx O(n^2ld)$ FLOPs, assuming $p \ll n$. In comparison, a vanilla $(s + l)$-head attention requires $O((s + l)n^2d)$ FLOPs, which is roughly $(s + l)/l$ times more than LS-attention.

During inference in a transformer model with auto-regressive generation, the KV-cache (Pope et al. (2023); Zhang et al. (2023)) is used to store the key and value vectors of previous tokens to compute the attention scores for the future queries in the MHSA operation. The size of the KV-cache for a full attention head grows linearly with sequence length. In contrast, it remains nearly constant for a local attention head. Therefore, if the total number of attention heads remains the same, LS-attention reduces the KV-cache size by a factor of approximately $(s+l)/l$ compared to MHSA during long-sequence generation.

5 EXPERIMENTAL RESULTS AND ANALYSIS

This section examines how sequence length affects logit explosion and training stability in transformer models for natural language and speech processing tasks, where local dependencies are typically dense. We further demonstrate that the composed attention mechanism, LS-Attention, mitigates logit explosion and training instability. Alternative structured self-attention mechanisms and other training stabilization methods are also evaluated on their applicability in addressing long-sequence training instability.

5.1 EXPERIMENTAL SETUP

Model Architecture For most experiments, we used the GPT-2 Small model with 12 layers, an embedding dimension (d) of 768, 12 attention heads, and a feedforward dimension of 3072. As the baseline multi-head self-attention (MHSA), we employed the CUDA implementation of Flash-attention, specifically Flash-attention-2 from Dao (2024).

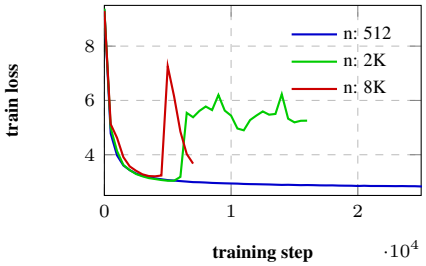
Hyperparameters of LS-Attention In experiments with LS-attention, we replaced the MHSA module with LS-attention. For an H-head LS-attention configuration, we used full (long-range) attention in one head, and local (short-range) attention in remaining $H - 1$ heads. The attention span for each local head was fixed at 100.

Training Details We trained all models using the AdamW optimizer with a weight decay of $1e{-}1$, $\beta_1 = 0.9$, and $\beta_2 = 0.95$. Gradient clipping was applied with a maximum norm of 1.0. The learning rate followed a cosine decay schedule with linear warmup: the maximum learning rate was set to $6e{-}4$, the minimum to $6e{-}5$, with $2K$ warmup steps and a total of $600K$ decay steps. Across all experiments, we fixed the total number of tokens per batch to 2^{19}. Consequently, when using longer sequence lengths, we proportionally reduced the number of sequences per batch to maintain a constant token budget. Unless stated otherwise, we used mixed-precision training with the bfloat16 (BF16) data type.

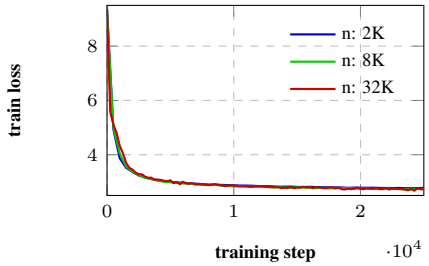
5.2 EXPERIMENTAL RESULTS ON NATURAL LANGUAGE DATA

Dataset and Preprocessing We conducted our experiments on the PG-19 dataset (Rae et al. (2020)), a collection of English-language books. All texts were normalized using NMT_NFKC and tokenized with a SentencePiece unigram model with a vocabulary size of $10K$.

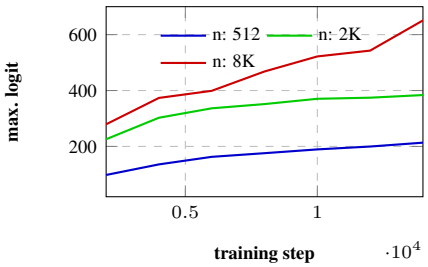
Results To investigate the effect of sequence length on training stability, we trained the above mentioned autoregressive baseline transformer (using Flash-attention as MHSA) for different sequence



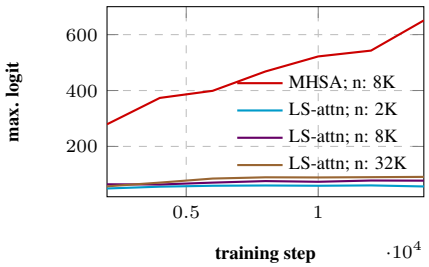
(a) Training curves of Flash-attention for sequence lengths $n = 512, 2K$, and $8K$.



(b) Training curves of LS-attention for sequence lengths $n = 2K, 8K$, and $32K$.



(c) Logit explosion of MHSA: the maximum logit increases with sequence length n.



(d) Comparison of logit explosion between MHSA and LS-attention.

Figure 3: Training stability and logit explosion on the PG-19 dataset (Rae et al. (2020)).

lengths. Figure 3a plots the training curves for sequence lengths of 512, $2K$, and $8K$. For the shorter sequence length ($n = 512$), the model trains stably, with the loss monotonically decreasing over the first $25K$ steps. However, when the sequence length is increased to $n = 2K$ and $n = 8K$, the training becomes unstable. In these cases, the loss initially decreases but then suddenly diverges, confirming that longer sequences can induce training instability. To further analyze this phenomenon, we tracked the maximum absolute logit values of the MHSA layes during training. As shown in Figure 3c, the maximum logit grows more rapidly with longer sequences. This result suggests that longer sequences cause greater logit explosion, which in turn contributes to the observed instability.

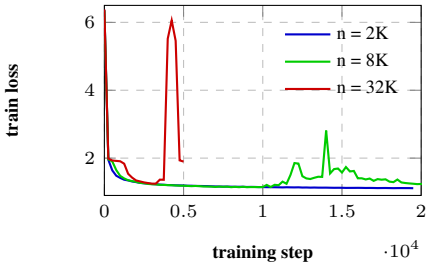
Next, we trained our baseline model by replacing the Flash-attention layers with LS-attention (comprising one full attention head and 11 local heads) for sequence lengths $n = 2K, 8K$, and $32K$. Figure 3b presents the training curves for these runs. As shown in the figure, the model with LS-attention does not exhibit any training instability during the first $25K$ training steps. To evaluate whether this improved training stability corresponds to mitigation of logit explosion, we compared the maximum absolute logit values of LS-attention and Flash-attention in Figure 3d. The figure clearly demonstrates that LS-attention substantially reduces the maximum logit values to negligible levels compared to Flash-attention, indicating that LS-attention effectively addresses logit explosion.

We provide additional results on the English split of the Wiki40B dataset (Dao et al. (2022)) in Appendix B. These results further confirm the above observations.

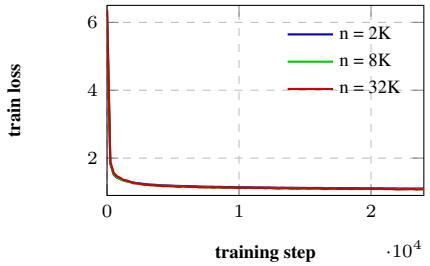
5.3 Experimental Results on Speech Data

Dataset and Preprocessing To validate our observations on a speech dataset, we used the $6K$-hour unlabeled subset of the LibriLight corpus (Kahn et al. (2020)). The audio data was tokenized using a pretrained HuBERT model (Hsu et al. (2021)) with a 500-cluster K-means tokenizer, resulting in a vocabulary size of 500. Our decoder only transformer models are autoregressively trained on the tokenized dataset.

Results Following the same protocol as with the PG-19 dataset, we trained the baseline model (using Flash-attention as the MHSA layers) on various sequence lengths. The training curves are



(a) Training curves of Flash-attention for sequence lengths $n = 2K, 8K$ and $32K$.



(b) Training curves of LS-attention for sequence lengths $n = 2K, 8K$ and $32K$.

Figure 4: Training stability evaluation on the LibriLight speech dataset ($6K$ split) Kahn et al. (2020).

plotted in Figure 4a. While the model trains stably with a sequence length of $2K$, it exhibits instability at longer lengths of $8K$ and $32K$. For these longer sequence lengths, the training loss diverges after an initial monotonic decrease. This instability correlates with logit explosion; Figure 8a in Appendix C shows that maximum absolute logit values increase more rapidly as sequence length grows.

We then repeated these experiments, replacing MHSA modules with LS-attention. As shown in Figure 4b, the model now trains stably even on sequences as long as $32K$. Furthermore, Figure 8b confirms that LS-attention mitigates the logit explosion seen in the baseline. These results indicate that LS-attention, which combines both local and full attention, effectively addresses the training instability encountered during long-sequence training.

5.4 Results with Larger Model

To assess whether the previous findings generalize to larger transformer models, we trained the **GPT-2 Large model** with 36 layers, 20 heads, embedding dimension 1280, and feedforward dimension 5120. The resulting model has approximately $750M$ parameters and was trained using the same setup described in Section 5.1. Figure 5 compares the training curves of the model using Flash-attention and LS-attention at a sequence length of $8K$. As before, the model with Flash-attention exhibits training instability, while LS-attention enables stable training.

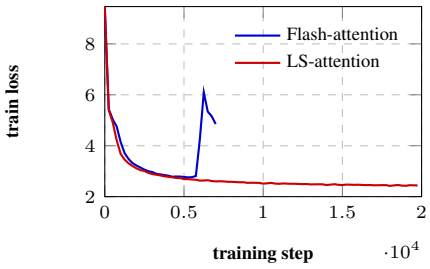


Figure 5: Comparison of training stability using larger model ($\approx 750M$ Parameters) on PG-19 dataset for sequence length $n = 8K$.

5.5 Comparison of Inference Time

We evaluated the inference latency of the LS-attention against the baseline using Flash-attention. Both models were benchmarked on an Nvidia A40 GPU with the BF16 data type. Latency was measured during a single forward pass over a batch of input sequences. The results, presented in Table 1, show that LS-sttention significantly reduces inference time on longer sequence lengths. At a sequence length of $n = 32K$, LS-attention is nearly 44% faster than the Flash-attention baseline. In general, as the sequence length increases, the time reduction achieved by LS-attention is expected to asymptotically approach a factor of H/l, where H denotes the total number of heads in Flash-attention and l represents the number of full-attention heads in LS-attention.

5.6 Additional Observations

Does Other Structured Self-attention Mitigates Long-Sequence Training Instability? Other structured self-attention mechanisms with local attention heads can also mitigate long-sequence train-

Seq. len (n)	Attention Type		Reduction
	Flash-attn	LS-attn	
$8K$	52.50	48.75	7.14%
$32K$	374.00	210	43.85%

Table 1: Reduction in inference time (in milliseconds per sequence) using LS-attention on Nvidia A40 GPU.

ing instability, as shown in Appendix D. However, while LS-attention can be efficiently implemented using publicly available software packages such as Flash Attention Dao et al. (2022); Dao (2024) and xFormers Lefaudeux et al. (2022), other attention mechanisms such as Guo et al. (2019) require hardware-specific custom implementations.

Does Existing Training Stabilation Methods Mitigates Long-Sequence Training Instability? In Appendix E, we evaluated three existing training stabilization methods: (1) QK-normalization (Henry et al. (2020)), (2) Z-loss (Chowdhery et al. (2023)), and (3) the AdaGC optimizer (Wang et al. (2025)). We found that Z-loss and AdaGC failed to mitigate long-sequence training instability, whereas QK-normalization led to significantly slower convergence than LS-attention – yielding more than 30% higher perplexity over the same number of training steps.

Can Flash-attention with Full Precision Training Mitigates Long-Sequence Training Instability? As discussed in Appendix F, Flash-attention with full FP32 precision training can mitigate long-sequence training instability. However, it requires over 15 times as many GPU hours to reach the same training loss as LS-attention.

How Sensitive Is LS-attention Performance to the Number of Local and Full Attention Heads? In our analysis in Appendix G, we find that LS-attention is only mildly sensitive to the number of both local and full attention heads, provided that at least one of each is present.

6 Conclusion

This paper identifies a key source of training instability in transformer models: the self-attention's limited ability to capture dense local dependencies. This limitation causes the logits of self-attention to explode when training on long sequences, leading to instability. To address this, we propose using a combination of local and full attention heads, where the local heads capture short-range dependencies and the full heads capture long-range dependencies. This composed self-attention, referred to as Long Short-attention (LS-attention), mitigates logit explosion and instability during long-sequence training. A wide range of experiments demonstrates that long-sequence training leads to logit explosion in standard multi-head self-attention, while LS-attention mitigates it, resulting in stable training. Furthermore, LS-Attention improves transformer efficiency compared to state-of-the-art multi-head self-attention implementations such as Flash-attention.

7 Broader Impact

Training instability in Transformer models poses a serious bottleneck in large-scale AI development, with substantial financial, operational, and environmental consequences. In both industrial and academic settings, a single failed training run – often occurring after days or weeks of computation – can result in thousands of wasted GPU-hours, escalating costs and carbon emissions. To mitigate this risk, practitioners resort to engineering-intensive workarounds such as curriculum-based sequence scaling and manual monitoring, further inflating resource overhead. By addressing a core source of instability, our approach offers a scalable and efficient path toward training robust, long-sequence Transformer models.

8 Reproducibility statement

Our source code is included in the supplementary materials.

REFERENCES

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 5723–5738. Association for Computational Linguistics, 2022.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020. URL https://arxiv.org/abs/2004.05150.

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Low-rank bottleneck in multi-head attention models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 864–873. PMLR, 2020.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11305–11315. IEEE, 2022.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1691–1703. PMLR, 2020.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *CoRR*, abs/1904.10509, 2019.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with io-awareness. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin Fathy

Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7480–7512. PMLR, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Alicia Golden, Samuel Hsia, Fei Sun, Bilge Acun, Basil Hosmer, Yejin Lee, Zachary DeVito, Jeff Johnson, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. Is flash attention stable? *CoRR*, abs/2405.02803, 2024.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In Helen Meng, Bo Xu, and Thomas Fang Zheng (eds.), *21st Annual Conference of the International Speech Communication Association, Interspeech 2020, Virtual Event, Shanghai, China, October 25-29, 2020*, pp. 5036–5040. ISCA, 2020.

Mandy Guo, Zihang Dai, Denny Vrandecic, and Rami Al-Rfou. Wiki-40b: Multilingual language model dataset. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pp. 2440–2452. European Language Resources Association, 2020.

Qipeng Guo, Xipeng Qiu, Xiangyang Xue, and Zheng Zhang. Low-rank and locality constrained self-attention for sequence modeling. *IEEE ACM Trans. Audio Speech Lang. Process.*, 27(12): 2213–2222, 2019.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 4246–4253. Association for Computational Linguistics, 2020.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:3451–3460, 2021.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7669–7673, 2020. `https://github.com/facebookresearch/libri-light`.

Akhil Kedia, Mohd Abbas Zaidi, Sushil Khyalia, Jungho Jung, Harshith Goka, and Haejun Lee. Transformers get stable: An end-to-end signal propagation theory for language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

B. Lefaudeux, F. Massa, D. Liskovich, W. Xiong, V. Caggiano, S. Naren, M. Xu, J. Hu, M. Tintore, S. Zhang, P. Labatut, D. Haziza, L. Wehrstedt, J. Reizenstein, and G. Sizov. xFormers: A modular and hackable transformer modelling library. `https://github.com/facebookresearch/xformers`, 2022.

Conglong Li, Minjia Zhang, and Yuxiong He. Curriculum learning: A regularization method for efficient and stable billion-scale GPT model pre-training. *CoRR*, abs/2108.06084, 2021.

Conglong Li, Minjia Zhang, and Yuxiong He. The stability-efficiency dilemma: Investigating sequence length warmup for training GPT models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 5747–5763. Association for Computational Linguistics, 2020.

Igor Molybog, Peter Albert, Moya Chen, Zachary DeVito, David Esiobu, Naman Goyal, Punit Singh Koura, Sharan Narang, Andrew Poulton, Ruan Silva, Binh Tang, Diana Liskovich, Puxin Xu, Yuchen Zhang, Melanie Kambadur, Stephen Roller, and Susan Zhang. A theory on adam instability in large-scale machine learning. *CoRR*, abs/2304.09871, 2023.

Kosuke Nishida, Kyosuke Nishida, and Kuniko Saito. Initialization of large language models via reparameterization to mitigate loss spikes. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 22699–22714. Association for Computational Linguistics, 2024.

Vittorio Pippi, Fabio Quattrini, Silvia Cascianelli, Alessio Tonioni, and Rita Cucchiara. Zero-shot styled text image generation, but make it autoregressive. *CoRR*, abs/2503.17074, 2025.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. In Dawn Song, Michael Carbin, and Tianqi Chen (eds.), *Proceedings of the Sixth Conference on Machine Learning and Systems, MLSys 2023, Miami, FL, USA, June 4-8, 2023*. mlsys.org, 2023.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018. URL `https://openai.com/research/language-unsupervised`.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019. URL `https://openai.com/research/language-unsupervised`.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Annual Conference on Neural Information Processing Systems 2017, USA*, pp. 5998–6008, 2017.

Guoxia Wang, Shuai Li, Congliang Chen, Jinle Zeng, Jiabin Yang, Tao Sun, Yanjun Ma, Dianhai Yu, and Li Shen. Adagc: Improving training stability for large language model pretraining. *CoRR*, abs/2502.11034, 2025.

Mitchell Wortsman, Peter J. Liu, Lechao Xiao, Katie E. Everett, Alexander A. Alemi, Ben Adlam, John D. Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-Dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs V: tuning large neural networks via zero-shot hyperparameter transfer. *CoRR*, abs/2203.03466, 2022.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Trans. Mach. Learn. Res.*, 2022, 2022.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M. Susskind. Stabilizing transformer training by preventing attention entropy collapse. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40770–40803. PMLR, 2023.

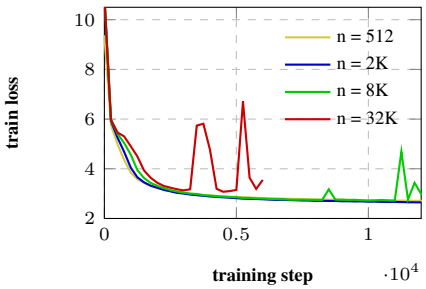
Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. H2O: heavy-hitter oracle for efficient generative inference of large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

A Use of Large Language Models

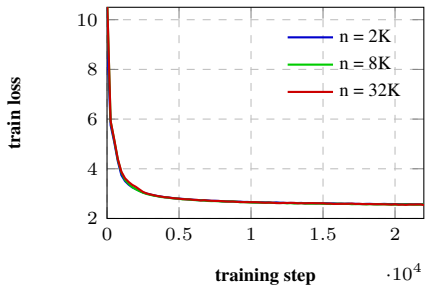
Large Language Models (LLMs) were used to improve textual clarity and to facilitate comprehensive information retrieval on a wide range of subjects from online resources.

B Results on Wiki40B (English split) Dataset

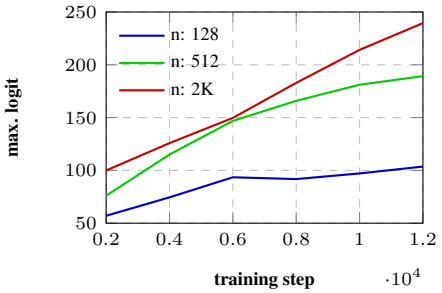
Dataset and Preprocessing To further validate our findings, we conducted experiments on the English split of Wiki40B (Guo et al. (2020)) dataset. This dataset is a cleaned version of Wikipedia designed for large-scale NLP tasks. Following the same preprocessing steps used for the PG-19 dataset, we normalized the texts with NMT_NFKC and tokenized them using a SentencePiece unigram model with a vocabulary size of $10K$.



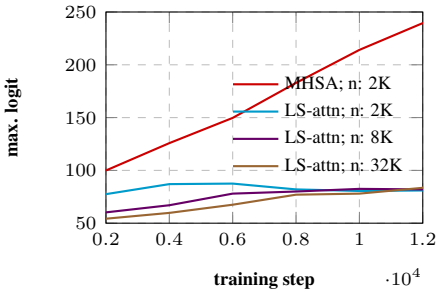
(a) Training curves of Flash-attention for sequence lengths $n = 512, 2K, 8K$ and $32K$.



(b) Training curves of LS-attention for sequence lengths $n = 2K, 8K$, and $32K$.



(c) Logit explosion of MHSA: the maximum logit increases with sequence length n.



(d) Comparison of logit explosion between MHSA and LS-attention.

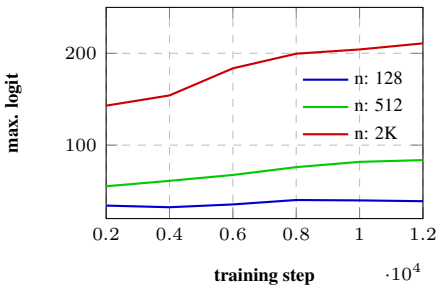
Figure 6: Training stability and logit explosion at long sequence lengths on the Wiki40b (English) dataset (Guo et al. (2020)).

Results As before, we trained the baseline transformer (using Flash-attention as the MHSA) with various sequence lengths. The training curves, shown in Figure 6a, mirror our previous findings. For shorter sequences (512 and $2K$), the model trains stably, with the loss decreasing monotonically over the first $15K$ steps. However, at longer sequence lengths ($8K$ and $32K$), the training becomes unstable and the loss diverges after an initial decrease. This instability again corresponds to a growth in logit values of MHSA. Figure 6c, which plots the maximum absolute logits of the MHSA layers, confirms that logit explosion is magnified by longer sequences, suggesting it is a primary cause of the training instability.

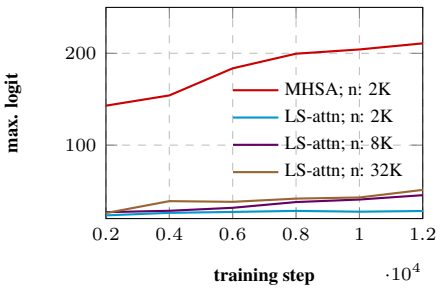
We then replaced the Flash-attention layers with the LS-attention and repeated the training for sequence lengths of $n = 2K, 8K$, and $32K$. The training curves in Figure 6b show that the model now trains stably across all tested lengths for the first $25K$ steps. Figure 6d compares LS-attention and Flash-attention with respect to logit explosion, confirming once again that the training stability achieved by LS-attention corresponds to effective mitigation of logit explosion.

C Logit Explosion Analysis for LibriLight Dataset

Building on the instability results for the LibriLight dataset (Section 5.3), we now analyze the underlying logit behavior. Figure 8a plots the maximum absolute logit values of the MHSA layers in the baseline transformer model. Consistent with our findings on other datasets, the logit values grow more rapidly as sequence length increases, reinforcing the link between sequence length and logit explosion. When compared to this baseline, LS-attention again proved to be effective. As shown in Figure 8b, LS-attention successfully mitigates the logit explosion observed in MHSA, reaffirming the trend seen across all the three datasets.



(a) Logit explosion of MHSA: the maximum logit increases with sequence length n.



(b) Comparison of logit explosion between MHSA and LS-attention.

Figure 8: Logit explosion at longer sequence lengths on LibriLight speech dataset (6K split).

D Evaluation of Alternative Structured Self-Attentions for Long Sequence Training

Several existing works (Child et al. (2019); Beltagy et al. (2020); Zaheer et al. (2020); Jiang et al. (2024); Guo et al. (2019)) have explored various structured self-attention mechanisms, including sparse attentions, to improve the computational efficiency of transformers on long sequences. Most of these structured self-attention methods also involve local attention and therefore should be able to stabilize long-sequence training. To verify this, we trained our baseline model with the MHSA layer replaced by the structured self-attention of Guo et al. (2019) on the PG-19 dataset with $n = 2K$. As expected, it was able to stabilize training. However, we found that the PyTorch implementation of this attention is nearly $2\times$ slower for sequence lengths below $8K$ and almost $10\times$ slower for sequence lengths of $32K$ compared to LS-Attention. The main advantage of LS-attention over the attention of Guo et al. (2019) is that LS-attention can be easily implemented using freely available packages such as Flash attention (Dao et al. (2022); Dao (2024)) and xFormers (Lefaudeux et al. (2022)). In contrast, optimized attention implementations, such as those in Guo et al. (2019), require hardware-specific programming.

E Evaluation of Existing Alternative Training Stabilization Methods

In this section, we evaluate whether existing alternative stabilization techniques can improve long-sequence training. To this end, we assessed three methods: (1) QK-normalization (Henry et al. (2020)), (2) Z-loss (Chowdhery et al. (2023)), and (3) the AdaGC optimizer (Wang et al. (2025)).

The learning curves for these methods, obtained by training our baseline model on the PG-19 dataset with a sequence length of $n = 2K$, are shown in Figure 9a. The results indicate that Z-loss and the AdaGC optimizer failed to stabilize training. Although QK-normalization successfully stabilized training, it converged too slowly compared to LS-attention, reaching a training loss of approximately 3.16 (i.e., perplixity 23.57) over the first $25K$ steps. In contrast, LS-attention achieved a substantially lower loss of about 2.77 (i.e., perplixity 15.96) over the same period, corresponding to roughly a 32% more reduction in perplexity.

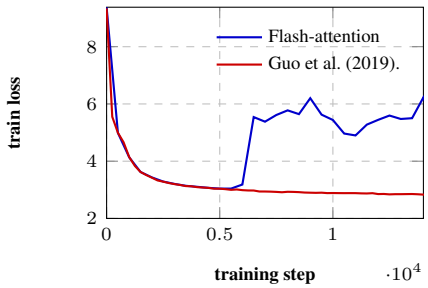
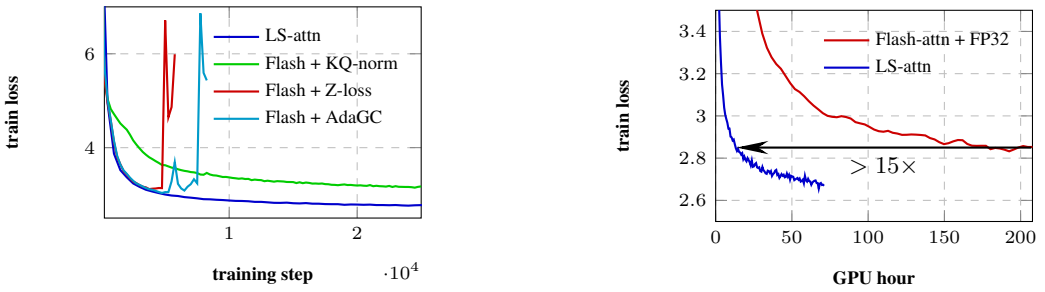


Figure 7: Evaluation of training stability by the structured attention of Guo et al. (2019) on PG-19 dataset with $n = 2K$.

F Comparison of LS-Attention (BF16) and FlashAttention (FP32)

In this section, we investigate whether training Flash-attention with full FP32 precision, rather than the default mixed precision with BF16, can stabilize long-sequence training. Prior work, such as Golden et al. (2024), has noted that Flash-attention – the efficient MHSA implementation – is particularly vulnerable to numerical instability caused by the reduced precision of low-bit datatypes. We therefore explored full-precision training of Flash-attention as a potential stabilization strategy.

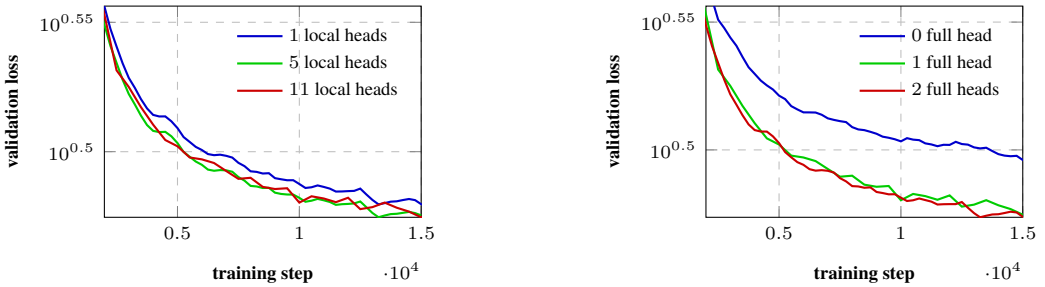
Our experimental results suggest that full-precision training can stabilize long-sequence training, though it incurs higher computational cost (both in terms of the number of training steps and GPU hours). Figure 9b compares the GPU hours required by LS-attention and full-precision Flash-attention to train the baseline model on the PG-19 dataset with $n = 8K$. The findings show that Flash-attention with full-precision training may require over 15 times more GPU hours to achieve comparable perplexity to LS-attention.



(a) Training loss vs. training steps.

(b) Training loss vs. GPU hours.

Figure 9: (a) Evaluation of alternative training stabilization methods for stabiling long sequence training. The following methods have been explored: (1) Flash-attention using QK-normalization, (2) Flash-attention with Z-loss, and (3) Flash-attention with AdaGC optimization. (b) Comparison of LS-attention and Flash-attention using full FP32 precision training. All the experiments have been conducted on PG-19 dataset.



(a) Impact of number of local heads.

(b) Impact of number of full attention heads.

Figure 10: Effect of the number of local and full attention heads in LS-attention. All experiments were conducted on the PG-19 dataset with $n = 8K$.

G Sensitivity Analysis of Hyperparameters

In this section, we analyze the sensitivity of LS-attention's performance to the number of local and full attention heads.

G.1 Sensitivity Analysis of the Number of Local Attention Heads

To investigate the effect of the number of local attention heads, we fixed the number of full-attention heads to 1 and varied the number of local attention heads to 1, 5, and 11. Note that setting the number of local attention heads below 11 while keeping one full-attention head reduces the total number of attention heads from the default value of 12, thereby decreasing the number of parameters in the self-attention layers. To compensate for this reduction, we increased the feed-forward dimension appropriately so that the total number of model parameters remained constant.

The learning curves for different numbers of local attention heads are shown in Figure 10a. The results suggest that varying the number of local attention heads has little effect on the convergence rate, with the configuration using a single local-attention head exhibiting slightly slower convergence.

G.2 Sensitivity Analysis of the Number of Full Attention Heads

To investigate the effect of the number of full-attention heads in LS-attention, we conducted experiments with the number of full-attention heads set to 0 (i.e., no full attention), 1, and 2, while keeping the total number of attention heads fixed at 12. The learning curves for these three settings are shown in Figure 10b. The figure suggests that increasing the number of full-attention heads from 0 to 1 yields a substantial performance improvement: the model with 0 full-attention heads reached a validation loss of 3.14 after $15K$ steps, whereas the model with 1 full-attention head achieved 2.98 over the same period. This improvement indicates that adding a single full-attention head helps the model capture long-range dependencies, thereby improving performance compared to the configuration without full attention. However, increasing the number of full-attention heads from 1 to 2 did not provide further benefits, suggesting that only a small number of full-attention heads is adequate for modeling long-range dependencies.