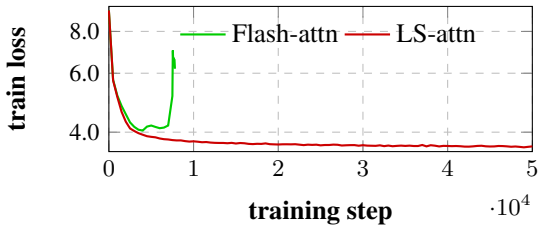
Short-Range Dependency Effects on Transformer Instability and a Decomposed Attention Solution

Anonymous ACL submission

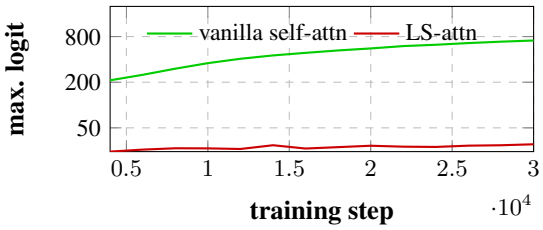
Abstract

Transformer language models have driven significant progress across various fields, including natural language processing and computer vision. A central component of these models is the self-attention (SA) mechanism, which learns rich vector representations of tokens by modeling their relationships with others in a sequence. However, despite extensive research, transformers continue to suffer from training instability – often manifesting as spikes or divergence in the training loss during a run.

In this work, we identify one source of this instability: SA's limited ability to capture short-range dependencies, especially in tasks like language modeling, where almost every token heavily relies on its nearby neighbors. This limitation causes the pre-softmax logits of SA to grow rapidly, destabilizing training. To address this, we propose decomposing the SA into local (short-range) and global (long-range) attention heads. This decomposed attention, referred to as Long Short-attention (LS-attention), mitigates logit explosion and results in more stable training compared to an equivalent multi-head self-attention (MHSA). Empirical comparisons with two alternative training stabilization methods show that LS-attention reduces the validation perplexity to nearly $2/5$ of that achieved by one method and reaches a similar perplexity as the other method using only $1/20$ of the GPU hours. Additionally, our experiments demonstrate that LS-attention reduces inference latency by up to 36% compared to a state-of-the-art implementation of equivalent MHSA.



(a) Plot of training loss over training steps.



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

Figure 1: Mitigation of training instability and logit explosion using LS-attention. The upper plots show that the training loss of an autoregressive transformer model with Flash-attention begins to diverge after some training steps, whereas the same model with LS-attention remains stable. The bottom plots compare the maximum absolute pre-softmax logits of vanilla MHSA and LS-attention during training. LS-attention prevents logit explosion by reducing the maximum logit magnitude to less than one-twentieth that of vanilla MHSA.

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. Their scalability and adaptability have made them the default choice for both academic research and industry-scale deployments. From BERT (Devlin et al., 2019) and GPT (Radford et al., 2018) to recent large-scale models, transformer language models have demonstrated exceptional abilities to model complex data distributions, generalize across tasks, and benefit from massive pretraining on unlabeled data.

Despite their 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 and 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 (Li et al., 2022, 2021) during training. Several studies (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 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 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 pre-softmax logits in SA 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 SA'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:

$$\mathrm{Y} = \mathrm{P}\mathrm{X}\mathrm{W}_v,$$

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: $\mathrm{P} = \mathrm{softmax}(\mathrm{S}) = \mathrm{softmax}(\mathrm{Q}\mathrm{K}^T) = \mathrm{softmax}(\mathrm{X}\mathrm{W}_Q\mathrm{W}_K^T\mathrm{X}^T)$[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 $\mathcal{O}(n^2)$ degrees of freedom. However, because P is derived from the product of two $n \times d$ matrices, its degree of freedom remains bounded above by nd. When $n \gg d$, this becomes a significant bottleneck. In tasks where all tokens depends on a small set of "keyword" tokens, the attention matrix becomes low-rank. However, in tasks requiring dense local dependencies – where nearly every token depends on its immediate neighbors – the attention matrix must be effectively high-rank. The inability of the low-rank structure to approximate such high-rank patterns forces the model to compensate by inflating the logits S, leading to instability during training.

Our 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 $\mathcal{O}(nl)$ degrees of freedom, where $l \ll n$ denotes the local window size. In contrast, global attention attempts to model interactions between all pairs of n tokens in the input sequence, requiring the representation of $\mathcal{O}(n^2)$ attention weights. This demand often exceeds the expressive capacity of the attention mechanism, since its parameterization is limited to $\mathcal{O}(nd)$ degrees of freedom. 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 $\mathcal{O}(nl')$, making it more compatible with the available degrees of freedom. Local attention is therefore more effective than global attention for capturing 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 decomposing the SA into local (short-range) and global (long-range) attention heads. This decomposed 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). A comparison with two alternative training stabilization methods shows that LS-attention either achieves significantly lower perplexity (as low as $2/5$ that of one method) or requires substantially fewer GPU hours (less than $1/20$) to reach comparable performance.

Efficiency of Our 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 global 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 global 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 LS-attention to be upto 36% more efficient during inference compared to Flash-attention (Dao et al., 2022;

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

Dao, 2024), the state-of-the-art efficient implementation of MHSA.

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

- We identify a key limitation of SA: its inability to model dense local dependencies in long sequences effectively. This limitation leads to logit explosion during training, contributing to instability in transformer models, particularly in tasks like language modeling.
- We propose Long Short-attention (LS-attention), which decomposes MHSA into long-range and short-range attention heads. Through extensive experimentation, we validate the effectiveness of LS-attention in mitigating logit explosion and training instability. Additionally, LS-attention offers improved computational efficiency on long sequences compared to vanilla MHSA.
- We empirically compare LS-attention with two alternative training stabilization methods. One method converges to a poor local optimum, with validation perplexity nearly $2.5\times$ higher than LS-attention after significant training progress. The other requires over $2.5\times$ more training steps and more than $20\times$ the GPU hours to achieve comparable performance.

2 Background

In this section, we provide a brief overview of the SA mechanism and introduce the notations used throughout this work.

Self-Attention For an input sequence $\mathrm{X} = [\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}]^T \in \mathbb{R}^{n\times d}$, where n is sequence length and d is the embedding dimension, the SA computes an output sequence $\mathrm{Y} = [\mathbf{y}_0, \ldots, \mathbf{y}_{n-1}]^T \in \mathbb{R}^{n\times d}$ such that $\mathbf{y}_i$ is a convex combination of the input tokens, i.e.,

$$\mathbf{y}_i = \sum_{j=0}^{n-1} \alpha_{ij}\mathbf{v}_j = \sum_{j=0}^{n-1} \alpha_{ij}\mathrm{W}_V\mathbf{x}_j \tag{1}$$

where $\mathbf{v}_j = \mathrm{W}_V\mathbf{x}_j \in \mathbb{R}^{d_v}$ is a value representation of token $\mathbf{x}_j$ and W_V is a learnable projection matrix. For each $i \in \{0, \ldots, n-1\}$, the set of weights $\{\alpha_{i0}, \ldots, \alpha_{i(n-1)}\}$ form a probability distribution. The weight α_{ij} determines the component of input token $\mathbf{x}_j$ in the output token $\mathbf{y}_i$, thus their dependency. In SA, the $\mathcal{O}(n^2)$ weights $\{\alpha_{ij}\}_{ij}$ are also learned as a pairwise function of the input tokens $\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}$. More precisely, α_{ij} is computed as

$$\alpha_{ij} = \frac{\exp\left(\mathbf{q}_i^T\mathbf{k}_j/\sqrt{d_k}\right)}{\sum_{j'=0}^{n-1}\exp\left(\mathbf{q}_i^T\mathbf{k}_{j'}/\sqrt{d_k}\right)} \tag{2}$$

where the query and key representation of each $\mathbf{x}_i$ is computed as $\mathbf{q}_i = \mathrm{W}_Q\mathbf{x}_i$ and $\mathbf{k}_i = \mathrm{W}_K\mathbf{x}_i$ with $\mathrm{W}_Q, \mathrm{W}_K \in \mathbb{R}^{d_k\times d}$ being two learnable projection matrices.

As stated in the introduction section, this operation can be written in matrix form as:

$$\mathrm{Y} = \mathrm{softmax}\left(\mathrm{QK}^T/\sqrt{d_k}\right)\mathrm{V}$$

where $\mathrm{Q}, \mathrm{K} \in \mathbb{R}^{n\times d_k}$ and $\mathrm{V} \in \mathbb{R}^{n\times d_v}$ are the matrices of all query, key, and value vectors.

Causal Attention In autoregressive decoding, the task is to predict the next token in a sequence given the preceding tokens. Formally, for an input sequence of tokens $\{x_0, \ldots, x_{n-1}\}$, each token x_i for $i = 1, \ldots, n-1$ is predicted based on the sequence $\{x_j\}_{j<i}$. In such tasks, the attention for each query token is restricted to tokens that come before it in the sequence. More precisely, each $\mathbf{y}_i$ is computed as

$$\mathbf{y}_i = \sum_{j\le i} \alpha_{ij}\mathrm{W}_V\mathbf{x}_j \tag{3}$$

where each α_{ij} is computed as in Eq. (2). Such attention is called causal attention.

Local Attention The above attention is sometimes referred to as global attention, as a query token can attend to another token across the full sequence (satisfying other restrictions like causality). It is also referred to as long-range attention because it can capture dependencies between tokens that are far apart. In contrast, local or short-range attention restricts each query token to attend only to its nearby neighbors. More precisely, in local attention, we compute $\mathbf{y}_i$ as

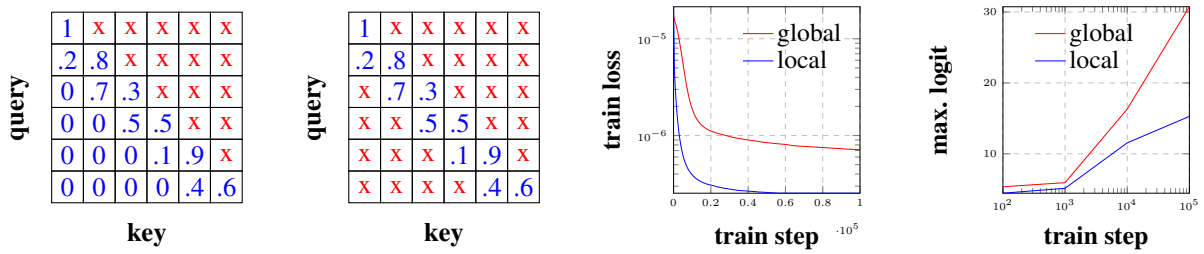
$$\mathbf{y}_i = \sum_{j=l}^{r} \alpha_{i(i+j)}\mathrm{W}_V\mathbf{x}_{i+j} \tag{4}$$

where each α_{ij} is computed as in Eq. (2) and l and r are small values compared to n. Therefore, in local attention each query token only attends to its nearby token within the range $\{i+l, \ldots, i+r\}$ For two sided local attention, l takes negative value and r takes a positive value. Local attention can be combined with causal attention by setting l to a small negative value and r to 0.

All SAs which restrict the attention to be computed to a subset $\mathcal{S} \subseteq \{(i,j) : 0 \le i, j \le n-1\}$ of all token pairs can be represented compactly by the following matrix notation

$$\mathrm{Y} = \mathrm{softmax}\left((\mathrm{QK}^T + \mathrm{M}_\mathcal{S})/\sqrt{d_k}\right)\mathrm{V} \tag{5}$$

where $\mathrm{M}_\mathcal{S}$ is a mask matrix defined as $\mathrm{M}_\mathcal{S}[i,j] = 0$ if $(i,j) \in \mathcal{S}$ and $-\infty$ otherwise. For example, in causal attention, $\mathcal{S} = \{(i,j) : j \le i\}$. For local causal attention with local attention span l, $\mathcal{S} = \{(i,j) : 0 \le i-j \le l\}$.



(a) Attention pattern for global attention.

(b) Attention pattern for local attention.

(c) Training curve for syn. experiment.

(d) Maximum logit during training on syn. experiment.

Figure 2: Comparison of representing dense local dependencies by local and global attention. (a) Global attention attempts to represent $\mathcal{O}(n^2)$ attention scores (shown in blue) using only $\mathcal{O}(nd)$ degrees of freedom. (b) Local attention focuses on $\mathcal{O}(nl')$ attention scores, where $l' \ll n$, making it a better fit for the available $\mathcal{O}(nd)$ capacity. (c) In a synthetic dense local dependency learning task, local attention achieves lower training loss. (d) Local attention is more resilient to logit explosion.

Multi-Head Attention Multi-head attention extends the self-attention mechanism by computing multiple attention operations in parallel, each with its own set of projection matrices. Specifically, given H heads, each head i computes:

$$\mathrm{Q}^{(i)} = \mathrm{X}\mathrm{W}_Q^{(i)}, \quad \mathrm{K}^{(i)} = \mathrm{X}\mathrm{W}_K^{(i)}, \quad \mathrm{V}^{(i)} = \mathrm{X}\mathrm{W}_V^{(i)}$$

where $\mathrm{W}_Q^{(i)}, \mathrm{W}_K^{(i)} \in \mathbb{R}^{d \times d_k}$, $\mathrm{W}_V^{(i)} \in \mathbb{R}^{d \times d_v}$ and typically $d_k = d_v = d/h$. Each head produces an output:

$$\mathrm{O}^{(i)} = \mathrm{softmax}\left(\mathrm{Q}^{(i)}{\mathrm{K}^{(i)}}^T/\sqrt{d_k}\right)\mathrm{V}^{(i)}.$$

The outputs from all heads are concatenated and projected back to the original dimensionality:

$$\mathrm{MHSA}(\mathrm{X}) = \mathrm{Concat}\left(\mathrm{O}^{(0)}, \ldots, \mathrm{O}^{(H-1)}\right)\mathrm{W}_O,$$

where $\mathrm{W}_O \in \mathbb{R}^{Hd_v \times d}$ is a learnable output projection matrix. Multi-head attention enables the model to jointly attend to information from different representation subspaces at different positions, which enhances the model's expressiveness.

3 Understanding the Limitation of Self-Attention

In this section, we analyze the ability of (global) self-attention to learn dense local dependensy. To this end, consider a causal next-token prediction task over sequences of length n, where the prediction of the next token depends only on the immediately preceding l tokens, with $l \ll n$. Let $\mathrm{Q} = [\mathbf{q}_0, \ldots, \mathbf{q}_{n-1}]^T \in \mathbb{R}^{n \times d}$ and $\mathrm{K} = [\mathbf{k}_0, \ldots, \mathbf{k}_{n-1}]^T \in \mathbb{R}^{n \times d}$ be the query and key matrices, where $\mathbf{q}_i$ and $\mathbf{k}_i$ denote the query and key vectors for the i-th token, respectively. For this task, the ideal attention matrix $\mathrm{P} \in \mathbb{R}^{n \times n}$ would satisfy $\mathrm{P}[i,j] > 0$ for $0 \le i - j \le l$, and $\mathrm{P}[i,j] = 0$ otherwise.

When attempting to learn this dependency pattern using causal (global) attention, the model aims to approximate a matrix P' such that $\mathrm{P}'[i,j] = \mathrm{P}[i,j]$ for $i - j \ge 0$, 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. As a result, it is difficult to find a low-rank parameterization that accurately captures this structure. During training, the attention mechanism attempts to replicate P' using $\mathrm{softmax}((\mathrm{QK}^T + \mathrm{M}_S)/\sqrt{d_k})$, but doing so requires representing $\mathcal{O}(n^2)$ non-masked entries in P' using only $\mathcal{O}(nd)$ degrees of freedom from QK^T. 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 $\{(i,j) : 0 \le i - j \le l'\}$, 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 $\mathcal{O}(nl')$ entries, which is significantly smaller than $\mathcal{O}(n^2)$ for global attention. As a result, local attention is better suited for learning dense local dependencies compared to global attention.

3.1 Validation through a Synthetic Task

Our synthetic task is designed to evaluate the representational power of the softmax operation, i.e., in capturing local dependencies when Q and K are allowed to freely take any values. The goal is to predict the output $\mathrm{O} = [\mathbf{o}_0, \ldots, \mathbf{o}_{n-1}]^T \in \mathbb{R}^{n \times d}$ of a sequence given the input $\mathrm{V} = [\mathbf{v}_0, \ldots, \mathbf{v}_{n-1}]^T \in \mathbb{R}^{n \times d}$, such that O satisfies $\mathrm{O} = \mathrm{PV}$ for a predefined 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 O from V

using an attention mechanism effectively requires learning Q and K such that $\mathrm{P} \approx \mathrm{softmax}((\mathrm{QK}^T + \mathrm{M}_{\mathcal{S}}))$ is satisfied, where $\mathrm{M}_{\mathcal{S}}$ denotes the appropriate masking matrix for global and local attention.

To that end, we generated a 2500×2500 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 $\mathbf{o}_i$ can be expressed as a unique linear combination of the $\mathbf{v}_j$s. This setup guarantees the uniqueness of P in the relation $\mathrm{O} = \mathrm{PV}$.

We trained both global and local attention operations for $100K$ steps using the Adam optimizer, with the key/query dimensionality d_k 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 global attention after $100K$ steps, indicating its superior ability to model dense local dependencies. Additionally, we tracked the maximum pre-softmax logit value (i.e., $||\mathrm{QK}^T||_\infty$) throughout training for both attention types, which is shown in Figure 2d. The figure reveals that while the logit values increase for both cases as training progresses, they rise much more sharply for global attention, indicating its higher susceptibility to the logit explosion problem when attempting to model local dependencies.

4 Long-Short Attention: Proposed Solution

As argued in the previous section, local attention mechanisms are more effective than global attention in modeling dense local dependencies. However, local attention cannot capture long-range dependencies. To address this limitation, our approach combines both local and global attention mechanisms to jointly model 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 and 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 attention patterns.

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

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

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 (global) attention heads. Therefore, the output of LS-attention is given by:

$$\text{LS-attn}(\mathrm{X}) = \mathrm{Concat}(\mathrm{O}^{(0)}, \ldots, \mathrm{O}^{(H-1)})\mathrm{W}_O$$

such that

$$\begin{aligned} \mathrm{O}^{(i)} &= \mathrm{softmax}\left((\mathrm{Q}^{(i)}{\mathrm{K}^{(i)}}^T + \mathrm{M}^{(i)})/\sqrt{d_k}\right)\mathrm{V}^{(i)} \\ &= \mathrm{softmax}\left((\mathrm{X}\mathrm{W}_Q^{(i)}{\mathrm{W}_\mathrm{K}^{(i)}}^T\mathrm{X}^T + \mathrm{M}^{(i)})/\sqrt{d_k}\right)\mathrm{X}\mathrm{W}_V^{(i)} \end{aligned}$$

where $H = s + l$ and $\mathrm{M}^{(i)}$ is the attention mask matrix for the i-th attention, and set to local attention mask for the first s heads and to the global attention mask for the last l heads. In practice, we do not implement the LS-attention using the above parallel form. Rather, we use the efficient SA implementation of (Dao et al., 2022; Dao, 2024; Shah et al., 2024).

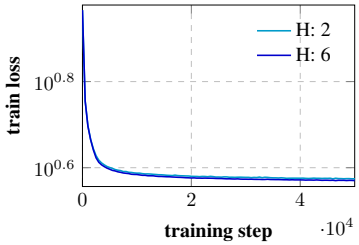
Runtime and Memory Requirements

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

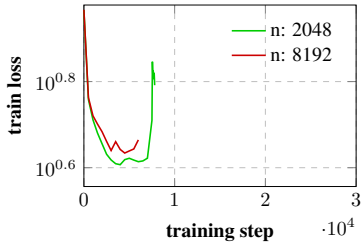
During inference in a transformer model with autoregressive generation, the KV-cache (Pope et al., 2023; Zhang et al., 2023) is used to store the key and value vectors of previous tokens to compute the attention scores for the future queries in the MHSA operation. The size of the KV-cache for a global 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

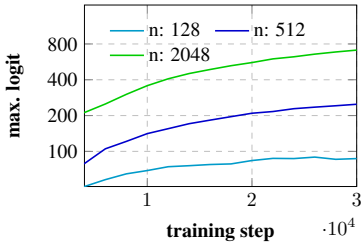
This section investigates the relationship between sequence length and the training instability of transformer



(a) Training curves with $H = 2$ and $H = 6$ at sequence length $n = 512$.

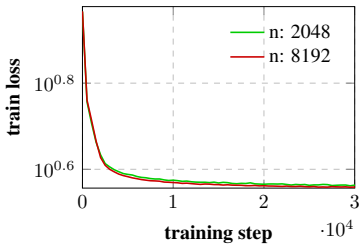


(b) Training curves for sequence lengths $n = 2048$ and $n = 8192$, with H set to 6.

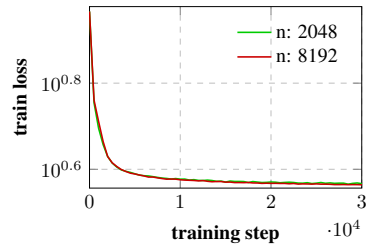


(c) Increase in maximum logit values during training for sequence lengths $n = 128$, 512, and 2048.

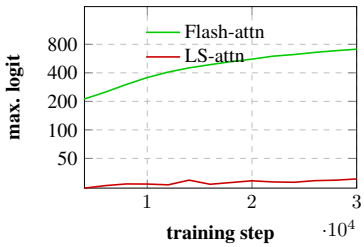
Figure 3: Training instability and logit explosion in Flash-attention at longer sequence lengths.



(a) Training curves for sequence lengths $n = 2048$ and $n = 8192$, with H set to 6.



(b) Training curves for sequence lengths $n = 2048$ and $n = 8192$, with H set to 2.



(c) Comparison of logit explotion in Flash-attn and LS-attn for sequence length $n = 2048$.

Figure 4: Mitigation of logit explotion and training instability using LS-attention.

models on a natural language modeling task, where local dependencies are typically dense. We also empirically validate the effectiveness of the proposed LS-attention in mitigating logit explosion and training instability. LS-attention is compared with two alternative methods for stabilizing transformer training. Finally, we compare the inference time of LS-attention with that of a state-of-the-art implementation of vanilla MHSA and provide an ablation study.

5.1 Experimental Setup

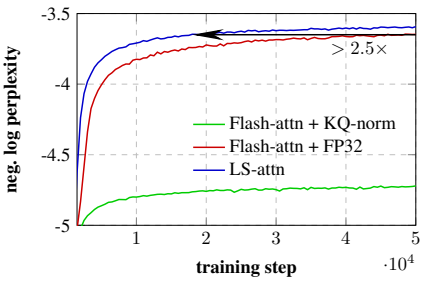
Model Architecture For experimental validation, we used a small-scale model with around $6.5M$ parameters. Our architecture is based on a GPT-2-style decoder, trained with an autoregressive loss. We set the number of layers to 6 and the embedding dimension d to 192. By default, the number of attention heads H was set to 6, and the inner dimension of the feedforward (FFN) layer, denoted by d_{ffn}, was set to $4d = 768$. In some experiments, we reduced the number of attention heads while keeping the per-head dimension fixed at $d/H = 32$. To maintain a similar number of parameters in these cases, we increased d_{ffn} accordingly (following Shazeer (2019); Ainslie et al. (2023)). As the baseline attention, we used the CUDA implementation of Flash-attention, specifically the FlashAttention2 implementation from Dao (2024).

Hyperparameters of LS-Attention In experiments with LS-attention, we replaced the MHSA module with our proposed LS-attention. For an H-head LS-attention configuration, one head was allocated for global (long-range) attention, while the remaining $H - 1$ heads were used for local (short-range) attention. The attention span for each local head was fixed at 50 for sequence lengths $n \leq 2048$ and 100 for longer sequences.

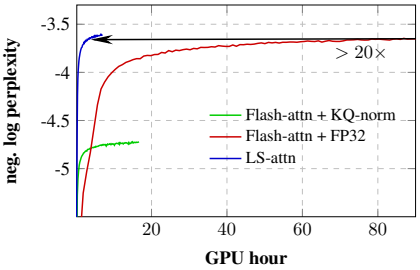
Optimization Hyperparameters We trained all models using the AdamW optimizer with a weight decay of 1e$-$1, $\beta_1 = 0.9$, and $\beta_2 = 0.95$. Gradient clipping was applied with a maximum norm of 1.0. The learning rate followed a cosine decay schedule with linear warmup: the maximum learning rate was set to 6e$-$4, the minimum to 6e$-$5, with 2000 warmup steps and a total of 600,000 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.

Dataset All experiments were conducted on the PG-19 dataset (Rae et al., 2020), which consists of full-length books. The dataset has a significantly high average document length, making it well-suited for evaluating long-range dependencies in language models. The text was normalized using NMT_NFKC normalization and tokenized using a SentencePiece tokenizer with a unigram model and a vocabulary size of $10K$.

All experiments were conducted on an NVIDIA A40 GPU. Unless stated otherwise, we used mixed-precision training with the bfloat16 (BF16) data type.



(a) Validation log perplexity vs. training steps.



(b) Validation log perplexity vs. GPU hours.

Figure 5: Performance comparison of LS-attention (in mixed BF16) with two alternatives: (1) Flash-attention trained with full FP32 precision, and (2) Flash-attention with QK-normalization (in mixed BF16). Sequence length n is set to 8192.

5.2 Investigation on Training Instability

In Section 3, we argued that longer sequence lengths lead to logit explosion in the self-attention layer, which in turn causes training instability in transformer language models. In this section, we investigate whether increasing the sequence length indeed causes such instability and logit explosion. We then evaluate whether LS-attention can mitigate this issue. To this end, we trained our baseline transformer model (which uses Flash-attention for its self-attention mechanism) with progressively longer sequence lengths. For smaller sequence lengths, such as $n = 512$, the model does not exhibit any signs of training instability – even when trained with a reduced number of attention heads H. As an illustration, Figure 3a shows the training curves for sequence length $n = 512$ with $H = 2$ and $H = 6$. It can be seen that the training loss decreases monotonically during the first $50K$ training steps. However, when the sequence length is increased to $n = 2048$ and $n = 8192$, the training becomes unstable – even for $H = 6$ – as shown in Figure 3b. For those sequence lengths, while the training loss initially decreases, it suddenly starts increasing as training progresses, clearly indicating that longer sequence lengths contribute to training instability in transformer models. To determine whether this instability is associated with logit explosion in the self-attention layer, we tracked the maximum absolute pre-softmax logit value during training for three different sequence lengths: $n = 128$, 512, and 2048. These are plotted in Figure 4c. The figure shows that the maximum logit value remains relatively small for $n = 128$, but grows significantly for $n = 2048$, suggesting that logit explosion contributes substantially to the observed training instability.

Next, we trained our transformer model using LS-attention as the self-attention module for sequence lengths $n = 2048$ and $n = 8192$ – the settings under which Flash-attention exhibited significant instability. Figure 4a shows the training curves for $H = 6$, where one head is global and the remaining five are local. As seen in the figure, the 6-head LS-attention does not exhibit any training instability during the first 50K training steps. To further assess the ability of LS-attention to mitigate training instability, we trained a model with only 2-head LS-attention – one global and one local – on the same longer sequence lengths ($n = 2048$ and $n = 8192$). The resulting training curves, shown in Figure 4b, indicate that even with just one global and one local head, LS-attention successfully stabilizes training for long sequences. To verify whether this training stability is accompanied by mitigation of logit explosion, we compared the maximum absolute pre-softmax logit values of LS-attention and vanilla self-attention in Figure 4c. The figure clearly demonstrates that LS-attention significantly reduces the maximum logit values to negligible levels compared to vanilla self-attention, suggesting that LS-attention effectively addresses the logit explosion.

5.3 Comparison with Alternative Training Stabilization Methods

We evaluated two alternative methods for stabilizing the training of our baseline model. First, we explored training with full FP32 precision instead of default mixed precision with BF16. Prior work, such as (Golden et al., 2024), has noted that Flash-attention – the efficient self-attention implementation – is particularly vulnerable to numerical instability due to the reduced precision of low-bit datatypes. Thus, training in full precision serves as a potential stabilization strategy. The second method we investigated is QK-normalization (Henry et al., 2020). Previous studies, including (Dehghani et al., 2023) and (Wortsman et al., 2024), have shown that QK-normalization can stabilize transformer training across various applications. The performance of our transformer model using these two alternative methods at sequence length $n = 8192$ is compared with LS-attention in Figure 5.

In Figure 5a, we plot the validation log perplexity over increasing training steps. The figure shows that both alternative methods overcome the training instability problem. However, Flash-attention with QK-normalization converges to a poor local optimum, achieving a validation perplexity of around 112.17 after the first $50K$ training steps – more than 2.5 times higher

than what the other two methods achieve after the same number of steps.

On the other hand, Flash-attention with full precision training is able to reach a validation perplexity of around 38.5 within the same $50K$ training steps. However, it requires larger than 2.5 times more training steps compared to LS-attention, which reaches similar perplexity in about $18,000$ training steps. Moreover, due to the full precision computations, each training step of Flash-attention with FP32 is significantly slower than LS-attention. As a result, it requires over 20 times the GPU hours to reach similar perplexity compared to LS-attention (as shown in Figure 5b).

5.4 Comparison of Inference Time

In this section, we compare the inference time of our transformer model with LS-attention to that of the same model using Flash-attention. Both configurations use the BF16 data type, and the total number of attention heads is set to 6. For LS-attention, the number of global heads is fixed at 1. Inference time were measured in batch processing mode, i.e., during the forward pass of a batch of input sequences through the model.

The result is presented in Table 1. The table shows that for sequence length $n = 2048$, replacing Flash-attention with LS-attention reduces the inference time by a modest 7.14%. However, when the sequence length is increased to $n = 8192$, the reduction improves significantly to 36.25%. This trend is expected: for longer sequences, the inference time becomes increasingly dominated by the cost of global self-attention heads, which scale quadratically with sequence length. Since LS-attention uses only one global head compared to six in Flash-attention, it becomes significantly more efficient at longer sequence lengths. As sequence length increases, the time reduction achieved by LS-attention is expected to asymptotically approach to a factor of H/l, where H is the total number of heads in Flash-attention and l is the number of global heads in LS-attention.

Seq. len (n)	Attention Type		Reduction
	Flash-attn	LS-attn	
2048	0.56	0.52	7.14%
8192	3.31	2.11	36.25%

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

5.5 Ablation Study

Vanilla MHSA uses only global attention heads, whereas LS-attention incorporates both local and global heads. In this section, we investigate whether the global heads in LS-attention significantly impact the model's performance. To this end, we compare the default LS-attention configuration (i.e., one global head and five local heads) with a variant that uses all six heads as local attention. The results, shown in Figure 6, suggest that the global attention head has a significant impact on performance. LS-attention with a global head achieves a validation perplexity of around 36, while the variant with no global head reaches only around 42 perplexity after the first $50K$ training steps. The results suggest that LS-attention effectively exploits both short-range and long-range dependencies by incorporating local and global attention.

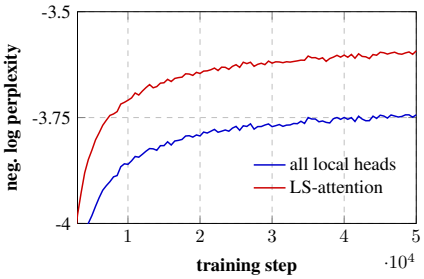


Figure 6: Impact of global attention head in LS-attention.

6 Conclusion

This paper identifies a key source of training instability in transformer models: the SA's limited ability to effectively model dense local dependencies. This limitation leads to the explosion of SA's pre-softmax logits, resulting in training instability for longer sequences. As a potential solution, we propose Long Short-attention (LS-attention), which decomposes standard MHSA into long-range (global) and short-range (local) attention heads. Extensive experiments validate LS-attention's ability to mitigate logit explosion and improve training stability. We compare LS-attention with two alternative training stabilization methods, demonstrating that LS-attention either achieves significantly better perplexity or converges faster – both in terms of training steps and GPU hours. Furthermore, LS-attention offers computational efficiency compared to state-of-the-art MHSA implementations.

7 Limitations

The limitation of self-attention (SA) in modeling dense local dependencies becomes particularly prominent when the sequence length n is significantly larger than the embedding dimension d. As a result, training instability in transformer models due to SA is more severe when training on longer sequences. Consequently, the benefits of LS-attention over standard MHSA – both in terms of training stability and computational efficiency – are more pronounced on longer sequences. In contrast, for shorter sequences, LS-attention may not offer significant advantages. Additionally, in applications where there is no dense local dependencies, incorporating local attention heads in LS-attention may not lead to performance improvements.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. GQA: training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4895–4901. Association for Computational Linguistics.

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. 2022. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5723–5738. Association for Computational Linguistics.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. 2022. 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*, pages 11305–11315. IEEE.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. 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*, pages 1691–1703. PMLR.

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, and 48 others. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

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

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with io-awareness. In *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*.

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, and 23 others. 2023. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 7480–7512. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186. Association for Computational Linguistics.

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. 2024. Is flash attention stable? *CoRR*, abs/2405.02803.

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

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. 2020. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4246–4253. Association for Computational Linguistics.

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

Akhil Kedia, Mohd Abbas Zaidi, Sushil Khyalia, Jungho Jung, Harshith Goka, and Haejun Lee. 2024. 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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations,*

ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

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

Conglong Li, Minjia Zhang, and Yuxiong He. 2022. The stability-efficiency dilemma: Investigating sequence length warmup for training GPT models. In *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*.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. 2020. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5747–5763. Association for Computational Linguistics.

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. 2023. A theory on adam instability in large-scale machine learning. *CoRR*, abs/2304.09871.

Kosuke Nishida, Kyosuke Nishida, and Kuniko Saito. 2024. Initialization of large language models via reparameterization to mitigate loss spikes. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 22699–22714. Association for Computational Linguistics.

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

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. 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.

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

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. *CoRR*, abs/1911.02150.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Annual Conference on Neural Information Processing Systems 2017, USA*, pages 5998–6008.

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

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. 2024. 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.

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

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. 2022. Scaling autoregressive models for content-rich text-to-image generation. *Trans. Mach. Learn. Res.*, 2022.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M. Susskind. 2023. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 40770–40803. PMLR.

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. 2023. H2O: heavy-hitter oracle for efficient generative inference of large language models. In *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.