Segment-Based Attention Masking for GPTs

Shahar Katz[*1] **Liran Ringel**[*2] **Yaniv Romano**[2,3] **Lior Wolf**[1]
[1]Blavatnik School of Computer Science, Tel Aviv University
[2]Department of Computer Science, Technion – Israel Institute of Technology
[3]Department of Electrical and Computer Engineering, Technion – Israel Institute of Technology
{shaharkatz3@mail,wolf@cs}.tau.ac.il, {liranringel@cs.,yromano@}technion.ac.il

Abstract

Causal masking is a fundamental component in Generative Pre-Trained Transformer (GPT) models, playing a crucial role during training. Although GPTs can process the entire user prompt at once, the causal masking is applied to all input tokens step-by-step, mimicking the generation process. This imposes an unnecessary constraint during the initial "prefill" phase when the model processes the input prompt and generates the internal representations before producing any output tokens. In this work, attention is masked based on the known block structure at the prefill phase, followed by the conventional token-by-token autoregressive process after that. For example, in a typical chat prompt, the system prompt is treated as one block, and the user prompt as the next one. Each of these is treated as a unit for the purpose of masking, such that the first tokens in each block can access the subsequent tokens in a non-causal manner. Then, the model answer is generated in the conventional causal manner. The Segment-by-Segment scheme entails no additional computational overhead. When integrated using a lightweight fine-tuning into already trained models such as Llama and Qwen, MAS quickly increases models' performances. Our code will be available at: https://github.com/shacharKZ/MAS-Segment-Based-Attention-Masking .

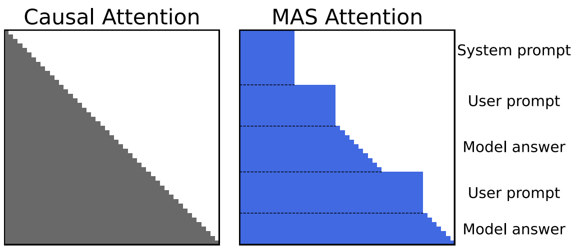

Figure 1: Causal vs. MAS attention: The plot shows binary values, with the y-axis as the current token index and the x-axis as tokens it can attend to. MAS masks prompts in blocks, enabling access to future tokens within the same block.

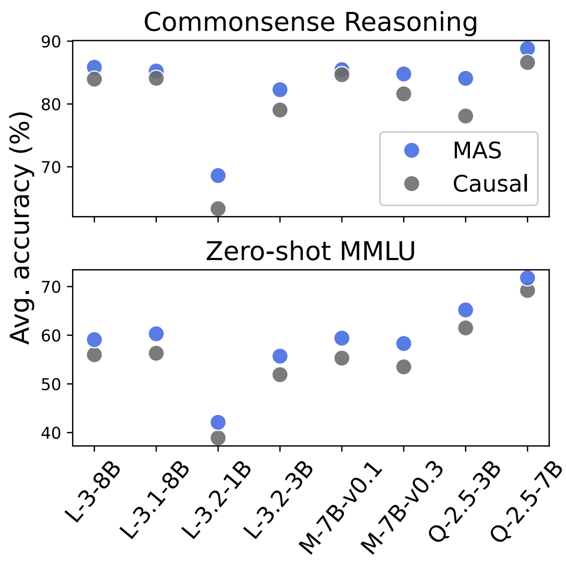

Figure 2: Fine-tuned models' performances on the Commonsense Reasoning benchmark and on zero-shot prompted version of MMLU. L,M,Q stand for Llama, Mistral and Qwen respectively.

1 Introduction

The introduction of the transformer architecture (Vaswani et al., 2017) has significantly advanced the field of natural language processing (NLP). Encoder transformer models (Devlin, 2018) read text bidirectionally, leveraging both preceding and subsequent words to build a rich contextual representation of the input. In contrast, decoder models, commonly referred to as GPT models (Radford et al., 2018), process text unidirectionally, from left to right. This unidirectional structure enables scalability and makes GPTs particularly effective for autoregressive tasks, such as conversational AI.

[*]Equal contribution.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 19308–19322

The original Transformer architecture introduced by Vaswani et al. (2017) utilized an encoder-decoder framework, where the encoder built a context for the input, and the decoder generated the output. However, this design requires approximately twice the number of parameters compared to decoder-only models with equivalent capacity. Efforts such as those by Dong et al. (2019), Raffel et al. (2020), and Tay et al. (2023) explored unified architectures, where a model's parameters are trained from scratch to function as both encoder and decoder. Despite their potential efficiency, these approaches failed to gain widespread adoption. In contrast, the remarkable success of decoder-only models, exemplified by GPT-3 (Brown et al., 2020), has shifted the field's focus toward architectures almost exclusively based on causal attention.

While the popularity of GPT models continues to grow, a key limitation is their inability to fully leverage information from future tokens. Moreover, the community's reliance on established GPT frameworks presents a significant challenge to adopting new attention mechanisms. For instance, solutions often build on existing models to gain traction, like chain-of-thought prompting (Wei et al., 2022), which facilitates iterative reasoning over previously processed tokens.

In this work, we introduce a novel approach called **Masked Attention by Segment (MAS)**, which leverages pre-trained GPT models and adjust them to utilize information from future tokens. MAS is inspired by the concept of unified architecture but **eliminates the need for training new architectures from scratch.** MAS achieves this by modifying the attention masking mechanism of GPTs, unmasking entire segments of input prompts. These segments appear as square blocks in Fig. 1. During autoregressive processing, the model reverts to using causal attention, selectively unmasking the attention between future and earlier tokens that are not available outside of training (the diagonal segments in the figure).

On a set of commonsense reasoning and summarization tasks, we show that our lightweight MAS adaptation can push publicly available GPTs to new state-of-the-art (SOTA) results; Fig. 2 illustrates these improvements for commonsense reasoning.

2 Related Work

Encoder-only models like BERT (Devlin, 2018) are primarily designed for bidirectional understanding tasks and excel in applications such as classification and question answering. While BERT can generate text autoregressively, each newly generated token changes the attention computation, requiring all dependent hidden states to be recomputed. Unlike decoder-only models, which use a key-value (KV) cache to efficiently generate multiple tokens during inference, BERT's bidirectional design prevents such optimization. This makes BERT impractical for token-by-token decoding.

The T5 framework (Raffel et al., 2020), with its encoder-decoder architecture, is effective for many NLP tasks due to its ability to incorporate bidirectional context during encoding and causal generation during decoding. However, SOTA and efficient performances in text generation are dominated by decoder-only models with causal masking, such as GPT-based architectures (Brown et al., 2020; Jiang et al., 2023; Yang et al., 2024; Dubey et al., 2024; Abdin et al., 2024). These models are highly efficient for token-by-token generation but cannot fully utilize input prompt information in their current design.

The most closely related work to our approach, PrefixLM, was explored in the T5 framework (Raffel et al., 2020). PrefixLM operates within a unified decoder-only architecture but enables bidirectional attention over a designated prefix of the input sequence while maintaining causal attention for the remainder. However, PrefixLM requires training from scratch and is limited to single-turn inputs, overlooking scenarios with multiple prefill phases, as often encountered in chat-based systems.

In contrast, our approach enables the easy enhancement of SOTA decoder-only models by unlocking the potential of bidirectional attention in non-generated segments through lightweight fine-tuning. Trained on massive corpora with causal masking, these models can be enhanced with limited hardware and just a few minutes of fine-tuning, quickly enabling them to effectively use bidirectional attention during the prefill phase.

Furthermore, MAS is tailored for chat-based tasks, where prompts are naturally segmented into components such as system instructions and user queries. In standard bidirectional attention, token hidden states depend on future tokens, making caching impractical. In contrast to PrefixLM, which processes the entire input as a single block, MAS separates the prompts so that the system prompt's KV cache remains independent of the varying user tokens and can be reused efficiently.

3 Background

Generative Pre-trained Transformer (GPT) models, also known as decoder-only transformers, are a family of language models (LMs) that generate each new token based on all previously produced tokens. The input, referred to as a *prompt*, is a sequence of n tokens denoted as $[t^1, \dots, t^n]$.

Processing begins with a **prefill** phase, in which the model consumes the entire prompt at once to predict the next token t^{n+1}. Afterwards, the model enters an **autoregressive decoding** phase, incrementally generating subsequent tokens t^{n+k} for $k = 2, 3, \dots$. At the k-th step, the input is the concatenation of the original prompt and all previously generated tokens:

$$[t^1, \dots, t^n, t^{n+1}, \dots, t^{n+k-1}]. \tag{1}$$

This autoregressive process continues until the model reaches a predefined generation limit or produces a designated stop token. The resulting sequence,

$$[t^1, \dots, t^{n+k}], \tag{2}$$

can then serve as a new prompt, initiating another cycle of prefill and decoding if needed.

3.1 Architecture

GPT architectures are built on a residual stream that connects multiple transformer blocks, each comprising three main components: a multi-head attention block (**Attn**), a multi-layer perceptron block (**MLP**), and layer normalization (**LN**) applied before each block.

The attention block operates using four parameter matrices: the query matrix $W_q \in \mathbb{R}^{d\times d}$, key matrix $W_k \in \mathbb{R}^{d\times d}$, value matrix $W_v \in \mathbb{R}^{d\times d}$, and output matrix $W_o \in \mathbb{R}^{d\times d}$. These matrices are divided into h heads, with each head using partitioned matrices of dimensions $\mathbb{R}^{d\times \frac{d}{h}}$. For the ℓ-th head, the projections are given as:

$$Q^\ell = XW_q^\ell, \tag{3}$$

$$K^\ell = XW_k^\ell, \tag{4}$$

$$V^\ell = XW_v^\ell, \tag{5}$$

where $Q^\ell, K^\ell, V^\ell \in \mathbb{R}^{n\times \frac{d}{h}}$ and $X = [x_1 \dots x_n] \in \mathbb{R}^{n\times d}$ represents the input sequence.

The attention scores for each head are computed using the position-encoded query and key by:

$$A^\ell = \text{softmax}\left(\frac{\tilde{Q}^\ell \tilde{K}^{\ell\top}}{\sqrt{d/h}} + M\right), \tag{6}$$

where $\tilde{Q}^\ell$ and $\tilde{K}^\ell$ are the query and key matrices after applying Rotary Position Embedding (RoPE) (Su et al., 2024).

Here, $M \in \mathbb{R}^{n\times n}$ enforces causal masking, ensuring that each token attends only to tokens that precede it in the sequence. This constraint is critical for maintaining the autoregressive property of GPT models, where predictions are conditioned only on prior tokens.

Each attention head output is computed as $A^\ell V^\ell$, and the outputs from all heads are concatenated and projected using W_o:

$$\text{Attn}(X) = [A^1V^1, \dots, A^hV^h]W_o. \tag{7}$$

Each attention layer is followed by an MLP layer. With the SwiGLU variant (Shazeer, 2020), the MLP architecture in the examined model uses three weight matrices: W_U, W_G, $W_D^\top \in \mathbb{R}^{d\times d_m}$, along with an activation function such as SiLU, f. The MLP output is computed as:

$$\text{MLP}(X) = (f(XW_U) \odot (XW_G))W_D \tag{8}$$

The output of the i-th transformer block, including layer normalization (LN), is computed as:

$$X^i_{\text{Attn}} = X^i + \text{Attn}(\text{LN}(X^i)), \tag{9}$$

$$X^{i+1} = X^i_{\text{Attn}} + \text{MLP}(\text{LN}(X^i_{\text{Attn}})). \tag{10}$$

The attention and MLP blocks complement each other: the attention mechanism captures dependencies between tokens across the sequence, while the MLP refines these representations independently at each token. Together, they iteratively enhance the hidden representation as it flows through the transformer layers, with layer normalization stabilizing the output at each step.

3.2 Fine Tuning

In this work, we examine publicly available GPTs from HuggingFace (Wolf et al., 2020), including Llama (Touvron et al., 2023), Qwen (Yang et al., 2024), and Mistral (Jiang et al., 2023). These models are pre-trained on extensive corpora to support general-purpose applications.

A common approach to enhance a GPT performance for specific tasks, such as reasoning or conversational applications, involves fine-tuning on task-specific datasets. However, full fine-tuning is computationally expensive. To address this challenge, adapter-based fine-tuning methods (Mangrulkar et al., 2022) have been developed, offering a computationally efficient alternative while maintaining comparable performance. Specifically, we focus on fine-tuning these general-purpose models using LoRA (Hu et al., 2022), the most widely adopted adapter-based fine-tuning technique.

LoRA (Low-Rank Adaptation) fine-tuning modifies only the **weight matrices**. Specifically, the original LoRA paper suggests to modify the attention mechanism only by fine-tuning the query matrix W_q and the value matrix W_v. Instead of directly updating the full matrices W_q and W_v, LoRA introduces low-rank decomposition matrices A and B such that the updated matrices become:

$$\tilde{W}_q = W_q + \Delta W_q = W_q + A_q B_q, \tag{11}$$

$$\tilde{W}_v = W_v + \Delta W_v = W_v + A_v B_v, \tag{12}$$

where $A_q, B_q^\top \in \mathbb{R}^{d\times r}$ and $A_v, B_v^\top \in \mathbb{R}^{d\times r}$, with $r \ll d$ being the rank of the decomposition. These low-rank matrices are trained during fine-tuning, while the original weights W_q and W_v remain frozen. This approach reduces the number of trainable parameters significantly, making fine-tuning more efficient.

By modifying only the query and value matrices with low-rank updates, LoRA enables efficient fine-tuning while preserving the pre-trained knowledge stored in the original model weights.

4 Method

Causal attention, as an autoregressive mechanism, restricts information flow to propagate only from earlier tokens to later ones. While this design is essential during the autoregressive decoding phase, it is unnecessarily restrictive during the prefill phase, in which the entire prompt is available at once. Specifically, causal masking prevents the model from leveraging information from later tokens in the prompt, introducing a constraint in the attention computation.

For instance, consider the following example from the commonsense reasoning task ARC-Challenge (Clark et al., 2018): *Please choose the correct answer to the question: Giant sloths lived in the Americas and the Caribbean during the Ice Age... Most of these sloths disappeared... some of these sloths lived alongside humans. What is the most likely reason that these last giant sloths became extinct? Answer1: disease... Answer4: humans as predators..."*

To correctly answer this question, the model must infer a specific information from the prompt (that human lived next to giant sloths) and ignore other (the fact that sloths lived through the Ice Age is a distracting detail). In a standard autoregressive model with causal masking, the text is read unidirectionally. This means the model cannot contextualize the final question while reading the initial sentences. Its success relies solely on its ability to memorize the prompt, as it cannot revisit earlier parts of the text when processing subsequent information. In contrast, humans readers can revisit earlier sentences or questions to focus on relevant details and build a coherent understanding.

The unidirectional nature of GPTs with causal masking imposes a limitation on their capabilities to integrate context from the entire prompt. To address this limitation, we propose Masked Attention by Segment (MAS), which adapts the attention mechanism to process full prompts more effectively. Similar to encoder-decoder, MAS operates in two modes – **i. Prefill Phase:** MAS removes the strict causal masking within each input prompt, allowing tokens to attend to both earlier and later tokens in the same block. **ii. Autoregressive Phase:** MAS reverts to standard causal masking, which ensures that each token only attends to preceding tokens. During training, this masking reflects the autoregressive nature of text generation, where future tokens are inaccessible at inference time.

Unlike previous approaches, MAS is designed for use with already trained GPT models, **eliminating the need to train new models from scratch.** To apply MAS, we modify the attention masking mechanism within general-purpose GPTs, and fine-tune the model on a dataset according to its downstream task. Specifically, the mask M in Eq. 6 is defined in the conventional GPT models as:

$$M_{i,j} = \begin{cases} 0 & \text{if } i \leq j, \\ -\infty & \text{if } i > j. \end{cases} \tag{13}$$

where $M \in \mathbb{R}^{n\times n}$ and $i, j \in \{1, 2, \ldots, n\}$ are token indices.

In MAS, the mask becomes

$$M_{i,j} = \begin{cases} 0 & \text{if } i \leq j \text{ or } S(i) = S(j), \\ -\infty & \text{otherwise.} \end{cases} \tag{14}$$

where $S(i)$ is a function that returns the segment ID of the token at position i. Tokens within the same prefill segment (such as a system prompt or user prompt) share the same segment ID.

By restricting modifications solely to the attention mask, MAS maintains computational efficiency and seamlessly integrates with existing architectures. In chat-based tasks, MAS leverages the structured nature of interactions, treating inputs as distinct blocks: the **system prompt** (providing initial instructions or context) and the **user prompt** (the user's actual input). These blocks are unmasked during the prefill phase, enabling tokens within the same block to attend to both earlier and later tokens. In contrast, the **assistant tokens** (model-generated response) use standard causal masking, ensuring that each token only attends to previously generated tokens.

During the training phase, given a set of chat-template examples, the prefill blocks are identified by special tokens that mark the beginning of the system and the user prompt segments. During test-time inference, MAS identifies the prompts as the inputs. When the model generates its response, it switches to causal masking.

5 Experiments

We evaluate MAS as an alternative to standard causal masking. In the following experiments, we fine-tune a set of GPT models, once using naive causal masking and once using MAS. To minimize computational costs, all fine-tuning setups are conducted with LoRA.

MAS fine-tuning: Our evaluation follows the experimental setup described in Liu et al. (2024) and Hu et al. (2023), where models are fine-tuned on a dataset containing 170,000 samples of reasoning tasks in the template of chat-based prompts. The training and test samples are from eight distinct datasets, each designed to evaluate different reasoning abilities. Some datasets, like BoolQ (Clark et al., 2019), provide naturally occurring yes/no questions about general knowledge, such as "Is your body temperature lower in the morning?". Others, such as WinoGrande (Sakaguchi et al., 2021), are fill-in-the-blank tasks with prompts like "It was easy for Amy but not Rachel to create a meal because [blank] had taken woodshop in school. Option 1: Amy, Option 2: Rachel". While couple datasets evaluate scientific reasoning, such as the ARC Challenge (Clark et al., 2018), others, like HellaSwag (Zellers et al., 2019), focus on narrative completion, and PIQA (Bisk et al., 2020) on physical commonsense reasoning.

All samples are structured as multiple-choice question-answering tasks and are written in an instruction-based format. The system prompt states: "Below is an instruction... Write a response that appropriately completes the request", followed by the user prompt, presenting the actual question, such as "Please answer the following question...".

This diverse range of tasks enables us to assess model performance across various reasoning abilities, offering a comprehensive evaluation of MAS's effectiveness in enhancing GPT-based models.

The hyperparameters used for all models are detailed in Appendix A. These settings are primarily based on the implementation used by Liu et al. (2024), with the exception of the batch size, which was adjusted due to computational constraints, and therefore also the learning rate. To ensure a fair evaluation, for each model we explored several learning rates in the range of $[1e-5, 1e-3]$. The learning rates selected for both masking approaches are the ones that achieved the highest average accuracy with causal masking, i.e., performance might be higher for MAS with other learning rates.

Our experiments include models from the Llama family (Dubey et al., 2024), Qwen 2.5 (Team, 2024), and Mistral (Jiang et al., 2023). For comparison, we also include results for GPT 3.5-turbo (Radford et al., 2018) when using a zero shot chain-of-thoughts (CoT), as provided by Hu et al. (2023).

The results in Table 1 illustrate that Masked Attention by Segment (MAS) consistently outperforms causal masking across most tasks and models. While the improvements are modest for some tasks and models, they are still significant: for example, MAS achieves an average accuracy increase of approximately 1.2% on BoolQ and PIQA. For the remaining tasks, the improvements range from 3.5% to 4.75%. These gains translate to an overall increase in average accuracy of around 1% to 7% across different models. When analyzing performance on individual tasks, MAS delivers better results for all models in nearly every case, as reflected in the MAS win rate shown in the last row of the table. The only exceptions are a single model that did not improve on BoolQ and the HellaSwag task, where MAS underperforms.

The decline in performance on HellaSwag, a task characterized by longer prompts, is attributed to our experimental setup: following previous work (Liu et al., 2024), we employed a maximum sequence length cutoff of 256 tokens during fine-tuning. To examine the fine-tuning-cutoff effect on MAS, we increased the maximum sequence length to 1024 tokens and retrained our models. Due to computational constraints, this setup was applied only to models with up to 3B parameters. The re-

Model	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
GPT 3.5-turbo CoT	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
Llama-3-8B	74.5	87.8	79.2	94.5	84.7	89.4	77.2	84.4	84.0
Llama-3-8B+MAS	74.9	88.6	81.5	93.2	88.0	92.3	81.5	86.8	**85.8**
Llama-3.1-8B	73.5	87.5	80.3	94.4	85.3	88.8	78.0	84.8	84.1
Llama-3.1-8B+MAS	74.8	88.0	82.0	91.6	86.0	92.2	81.7	85.8	**85.2**
Llama-3.2-1B	62.1	72.1	71.0	58.7	65.4	67.0	49.3	61.0	63.3
Llama-3.2-1B+MAS	62.4	78.2	73.0	69.1	70.2	72.7	55.5	67.6	**68.6**
Llama-3.2-3B	70.0	83.4	77.2	90.7	79.5	83.0	70.6	78.0	79.0
Llama-3.2-3B+MAS	71.1	86.0	79.6	90.6	83.9	89.4	75.1	82.6	**82.3**
Mistral-7B-v0.1	74.3	88.4	80.0	94.8	85.6	88.8	78.8	86.6	84.7
Mistral-7B-v0.1+MAS	70.9	88.5	82.5	91.4	88.2	92.6	80.5	89.2	**85.5**
Mistral-7B-v0.3	74.7	87.6	79.8	74.5	84.2	88.6	76.8	86.6	81.6
Mistral-7B-v0.3+MAS	74.6	89.1	81.0	89.6	88.2	91.0	78.7	86.2	**84.8**
Qwen2.5-3B	60.4	84.7	75.9	66.7	77.4	92.8	81.7	85.0	78.1
Qwen2.5-3B+MAS	68.3	85.6	80.6	90.7	83.8	93.5	84.0	86.2	**84.1**
Qwen2.5-7B	72.2	90.1	79.4	94.4	83.3	95.2	87.6	90.6	86.6
Qwen2.5-7B+MAS	73.7	90.3	83.5	94.5	88.8	96.7	90.1	93.0	**88.8**
MAS win rate	75.0	100.0	100.0	50.0	100.0	100.0	100.0	87.5	100.0

Table 1: Results on the Commonsense Reasoning benchmark.

model	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
Llama-3.2-1B	62.2	74.0	70.3	73.8	64.9	67.6	50.1	62.4	65.6
Llama-3.2-1B+MAS	64.5	80.0	74.7	87.2	74.2	78.8	61.3	75.2	74.5
Llama-3.2-3B	70.2	84.0	78.1	92.5	79.9	83.0	69.7	78.8	79.5
Llama-3.2-3B+MAS	73.8	87.4	81.5	94.3	85.5	90.7	77.4	84.4	84.4
Qwen2.5-3B	52.0	84.1	78.4	89.5	77.4	92.0	79.9	86.2	79.9
Qwen2.5-3B+MAS	72.0	87.2	80.8	93.2	84.8	93.6	82.2	87.6	85.2
MAS win rate	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Table 2: Results of Commonsense Reasoning Tasks when training with sequence length cutoff of 1024.

model	train → test	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
1B	MAS → CA	62.2	50.3	31.7	23.4	52.8	18.6	22.4	20.0	35.1
	CA → MAS	0.0	0.0	0.1	0	0.0	0.0	0.0	0.0	0.0
3B	MAS → CA	62.4	78.2	68.8	22.6	71.0	81.1	63.3	68.6	64.4
	CA → MAS	19.6	2.4	5.9	0.1	0.9	5.3	4.7	5.6	5.5

Table 3: Average accuracy on the Commonsense Reasoning tasks for different attention mechanisms while training and evaluating Llama-3.2 . CA stands for causal attention. The results for MAS → MAS and CA → CA can be found in Tab. 1 above.

sults, presented in Table 2, indicate that a longer context window enhances HellaSwag performance for both MAS and the causal masking baseline. Notably, MAS-enhanced models now outperform causal masking across all evaluated tasks, leading to an increase in overall average scores.

Overall, these findings suggest that MAS has the potential to enhance the performance of downstream tasks of pretrained LLMs, which were originally trained using causal masking.

MAS fast adaptation: To check whether adapting to the MAS attention scheme requires more training than the baseline fine-tuning, we train Llama-3.2 for 3 epochs (63,000 steps), with checkpoints saved every one-third of an epoch (7,000 steps). Figure 3 presents the average accuracy at each checkpoint. Evidently, MAS consistently outperforms causal masking at every stage of training.

During the initial epoch, the accuracy gap between MAS and causal masking grows steadily, with MAS showing incremental improvements over causal masking. After the first epoch, performance gains slow for both methods, as training approaches a plateau. However, the performance gap established during the early stages remains consistent throughout the remaining training steps.

These findings highlight two key points: i. Rapid adaptation: MAS achieves significant performance improvements early in training, and ii. Sustained progress: MAS maintains a stable advantage throughout training, with consistent improvement and no sharp fluctuations between steps.

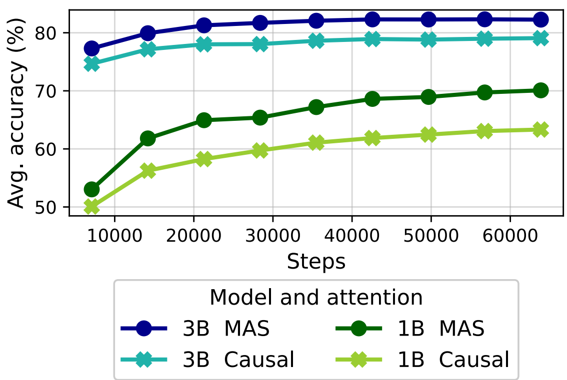

Figure 3: The average accuracy on the Commonsense Reasoning during the fine-tuning of Llama-3.2, either 1B or 3B using conventional (causal) attention or MAS.

Fine-tuning impact: Next we examine whether the fine-tuning is essential for MAS, and can MAS fine-tuned models perform well using only causal masking. The results, shown in Table 3, reveal that models fine-tuned with causal masking but evaluated with MAS perform poorly, emphasizing the necessity of fine-tuning for adapting to MAS. In contrast, models fine-tuned with MAS and evaluated with causal masking achieve moderate performance, approximately half the accuracy obtained when evaluated with MAS, further emphasizing the critical role of MAS-specific fine-tuning for optimal performance.

We also present an ablation study in Appendix B examining the effects of training different subsets of model parameters, such as parts of the attention matrices and the MLP layers. Additionally, Appendix C presents an ablation study on the hyperparameters involved in the fine-tuning process.

Zero-shot: To further validate MAS, we evaluate the MAS-finetuned models in a zero-shot setting on the MMLU benchmark (Hendrycks et al., 2021) without additional training. MMLU comprises 57 sub-tasks spanning commonsense reasoning, general knowledge, and logic, offering a comprehensive evaluation of LM performance. We use the same chat-based prompt templates from our commonsense tasks (Hu et al., 2023) to prompt MMLU. The results in Table 4 show that all models achieve an average accuracy improvement of approximately 3%, demonstrating that the benefits of MAS transfer effectively across diverse domains.

Model	Causal	MAS
Llama-3-8B	56.0	59.1
Llama-3.1-8B	56.3	60.3
Llama-3.2-1B	38.9	42.1
Llama-3.2-3B	51.9	55.7
Mistral-7B-v0.1	55.3	59.4
Mistral-7B-v0.3	53.5	58.3
Qwen2.5-3B	61.5	65.2
Qwen2.5-7B	69.2	71.8

Table 4: Zero-shot evaluation – average accuracy of fine-tuned models on the 57 MMLU tasks.

Additional summarization task: To demonstrate the applicability of MAS in another type of generation tasks, we evaluated it on three standard summarization benchmarks: BillSum (Kornilova and Eidelman, 2019), XSum (Narayan et al., 2018), and CNN/DailyMail (Hermann et al., 2015). For each dataset, models with up to 3B parameters were fine-tuned with both MAS and standard causal masking, and evaluated using ROUGE-L (Lin, 2004) and METEOR (Banerjee and Lavie, 2005). Due to computational constraints, input samples were limited to a maximum of 2048 tokens, with up to 50,000 training examples per dataset, and evaluation performed on approximately 5,000 test samples. BillSum has fewer samples of maximum 2048 token length; therefore, it was trained and tested on only 10,000 and 1800 samples, respectively. As shown in Table 5, MAS outperforms causal masking in most settings across datasets and model sizes. These results indicate that the benefits of MAS extend beyond generating answers for question-answering tasks and also improve performance on other type of generative tasks, such as summarization.

Emergent attention patterns: To gain insights into how MAS influences model behavior, we **visualize attention maps** of models fine-tuned with

Model Name	BillSum		XSum		CNN-DM	
	ROUGE-L	METEOR	ROUGE-L	METEOR	ROUGE-L	METEOR
Llama-3.2-1B	0.42951	0.42302	0.28963	0.30218	0.28032	0.35390
Llama-3.2-1B+MAS	0.45337	0.44531	0.30875	0.32312	0.29337	0.35016
Llama-3.2-3B	0.46416	0.46867	0.33351	0.35109	0.29693	0.36297
Llama-3.2-3B+MAS	0.48238	0.48383	0.34493	0.36598	0.30716	0.36958
Qwen2.5-3B	0.45502	0.46479	0.30025	0.31552	0.28961	0.35287
Qwen2.5-3B+MAS	0.47364	0.46799	0.32571	0.34207	0.30048	0.36019

Table 5: Summarization tasks' performances for models fine-tuned with MAS and causal masking.

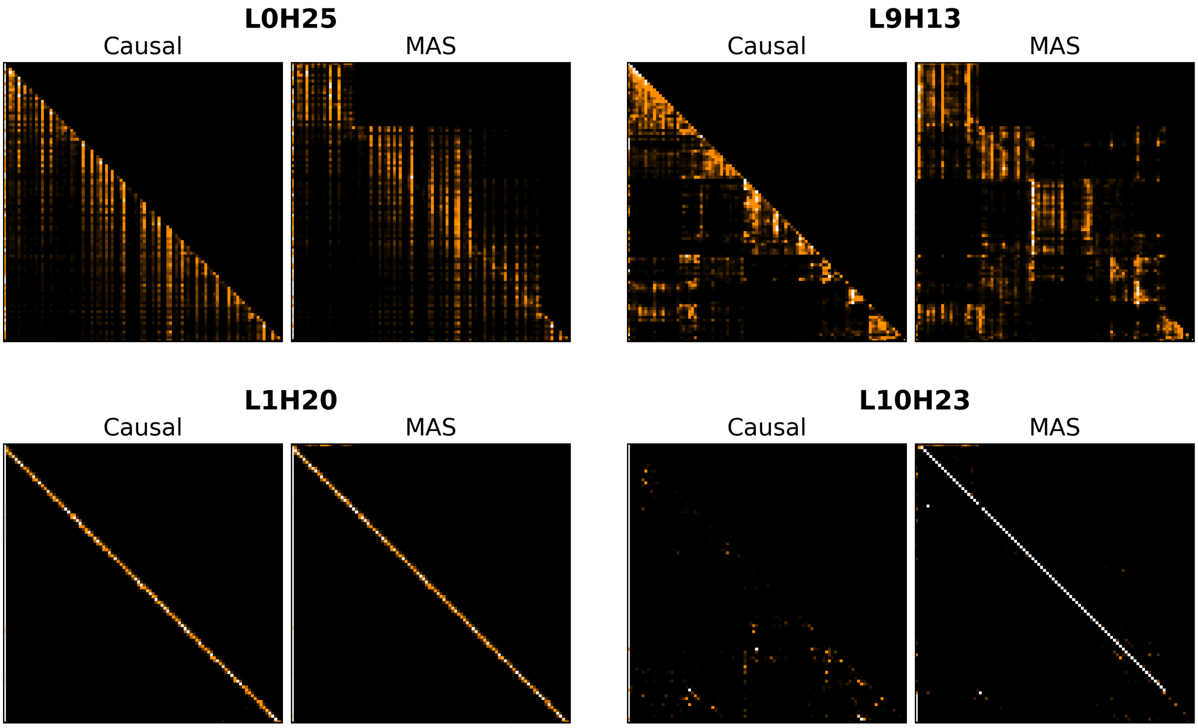

Figure 4: The attention maps of two instances of Llama-3.2-1B fine-tuned for commonsense reasoning tasks, one with standard causal masking and the other with MAS, reveal distinct patterns: (i) L1H25 (layer 0, head 25) exhibits **N-gram Patterns**, indicating attention focused on sequences of consecutive tokens. (ii) L9H13 demonstrates **Block-Specific Patterns**, concentrated within defined blocks of the prompts. (iii) L10H23 showcases a **Forward-Looking** behavior, attending precisely to the next tokens within the same block. (iv) L1H20 serves as an example of **Preserved Patterns**.

MAS. In models trained with standard causal masking, attention maps exhibit a lower triangular structure, reflecting the restriction that each token can only attend to preceding tokens. In contrast, MAS introduces a block-based structure in the attention maps, allowing tokens within the same block to attend to each other.

Figure 4 provides a comparison between two instances of Llama-3.2-1B, as detailed in section 5: one fine-tuned with standard causal masking and the other fine-tuned with MAS. The input prompt in this comparison consists of three distinct components: the system prompt, the user prompt, and the assistant's output. In our analysis, we identified four distinct attention patterns emerging across MAS attention heads:

- **Preserved Patterns:** Certain heads maintain their original behavior from standard causal masking, attending primarily to preceding tokens.
- **Block-Specific Patterns:** A frequently observed pattern in attention maps is the presence of vertical lines indicating specific tokens within the prompt to which many other tokens attend. In causal attention, these vertical lines are confined below the

main diagonal, but in MAS, they extend above the diagonal or even span the height of an entire block. Occasionally, this pattern highlights the boundaries of a block by emphasizing its first or last token.

- **N-gram Patterns:** This pattern appears as short crossed vertical or horizontal lines over the main diagonal, typically, spanning only a few tokens. It suggests MAS's ability to identify the context of a specific token using its neighboring tokens, including those preceding and following it. While causal attention enables context identification from preceding tokens, MAS extends this capability to include future tokens as well.

- **Forward-Looking Patterns:** A Certain attention heads focus on tokens located exactly a few steps ahead, identifiable by cross secondary diagonals above the main diagonal.

These patterns are illustrated in Figure 4, with a particularly notable example of the forward-looking pattern: in layer 10, head 23, an almost perfect diagonal line appears just above the main diagonal of the attention matrix. This behavior demonstrates the ability of certain heads to anticipate and focus on subsequent tokens within the same block. Importantly, this pattern aligns with MAS's design, as it breaks between the system and user prompts, respecting the separation of blocks. This forward-looking attention terminates at the start of the assistant's output, where causal masking resumes during the autoregressive phase to ensure proper token-by-token generation.

Additional results are provided in Appendix D.

6 Conclusions

In this work, we introduced Masked Attention by Segment (MAS), a novel method to enhance the performance of GPTs, already trained decoder-only LMs, by enabling them to leverage future tokens during input processing. This approach addresses the inherent limitations of standard causal attention through a straightforward fine-tuning process. Extensive experiments on commonsense reasoning benchmarks demonstrated the scalability and effectiveness of MAS across diverse tasks and training setups. Notably, MAS consistently outperformed causal masking in accuracy, with improvements evident early in training and maintaining a stable advantage as training progressed.

Limitations

While Masked Attention by Segment (MAS) offers a promising enhancement for GPT-based models, several limitations warrant consideration.

First, MAS requires fine-tuning existing models. Although this process is significantly less resource-intensive than training models from scratch, it still imposes computational and time constraints, which may limit its accessibility for broader adoption.

For evaluation, we selected models from today's most capable and widespread publicly available GPTs, such as Llama-3.2 and Qwen2.5, focusing on their base versions rather than their instruction-tuned counterparts. This choice aligns with previous work (Liu et al., 2024), which used base models to avoid the complexities introduced by model-specific templated prompts. While this approach provides a consistent evaluation framework, we acknowledge that MAS may behave differently on other models or specialized setups.

We run experiments on commonsense reasoning benchmarks, drawn from prior studies (Hu et al., 2023; Liu et al., 2024), and a wide array of generative Q&A tasks as part of the MMLU benchmarks. Future research will investigate MAS's applicability to additional tasks such as text summarization and translation.

Ethics Statement

This work aims to enhance language models by introducing a novel method to improve their performance through advancements in the attention mechanism. We recognize the potential of such technologies and emphasize the importance of their responsible use. While our contributions are intended to support the development of more aligned models, we stress the need of preventing misuse, such as generating harmful content. Future research should focus on promoting applications that align with societal benefit.

Acknowledgements

This work was supported by a Tel Aviv University Center for AI and Data Science (TAD) grant. This research was also supported by the Ministry of Innovation, Science & Technology, Israel (1001576154) and the Michael J. Fox Foundation (MJFF-022407). L. R. and Y. R. were supported by the European Union (ERC, SafetyBounds, 101163414). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This research was also partially supported by the Israel Science Foundation (ISF grant 729/21). Y. R. acknowledges additional support from the Career Advancement Fellowship at the Technion. The contribution of S.K. is part of a PhD thesis research conducted at Tel Aviv University. The contribution of L.R. is part of a PhD thesis research conducted at Technion University.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

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

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5276.

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

Anastassia Kornilova and Vlad Eidelman. 2019. Billsum: A corpus for automatic summarization of us legislation. *arXiv preprint arXiv:1910.00523*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Shih-yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin

Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

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

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

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

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Attention heads of large language models: A survey. *arXiv preprint arXiv:2409.03752*.

Hyperparameters	Value
LoRA Rank	32
LoRA alpha	64
Dropout	0.05
Optimizer	AdamW
LR scheduler	Linear
Batch size	8
Cutoff length	256
Warmup steps	100
Epochs	3

Table 6: Common hyperparameters for Commonsense Reasoning fine-tuning

Model	$\mathbf{LR}_{qv}$	$\mathbf{LR}_{Ex}$	Data type
Llama-3-8B	2e-4	1e-4	FP16
Llama-3.1-8B	2e-4	2e-4	FP16
Llama-3.2-1B	2e-4	2e-4	FP16
Llama-3.2-3B	8e-5	2e-4	FP16
Mistral-7B-v0.1	8e-5	3e-5	FP16
Mistral-7B-v0.3	1e-4	1e-4	FP16
Qwen2.5-3B	1e-4	2e-4	BF16
Qwen2.5-7B	1e-4	3e-4	BF16

Table 7: Model specific hyperparameters for Commonsense Reasoning fine-tuning

A Additional Fine-Tuning Implementation Details

In section 5, we presented results for the Commonsense Reasoning tasks. Here, we provide additional implementation details.

A.1 Tasks

The benchmarks and their implementations are based on Hu et al. (2023). The evaluation includes eight distinct tasks: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), OBQA (Mihaylov et al., 2018), ARC-e and ARC-c (Clark et al., 2018). The fine-tuning dataset, Commonsense170K, consists of training samples from these benchmarks, formatted into structured templates that include a system prompt, additional input, and an answer for the model to complete.

A.2 Hyperparameters

The hyperparameters used for the main fine-tuning in section 5 are detailed in Table 6. Model-specific hyperparameters are provided in Table 7, where $\mathbf{LR}_{qv}$ refers to the setup in which only the W_q, W_v matrices are trained, and $\mathbf{LR}_{Ex}$ corresponds to the configuration that includes training additional matrices (W_q, W_k, W_v, W_U, W_D). For both causal and MAS we used the same hyperparameters. We adjusted the learning rate between the two setups of $\mathbf{LR}_{qv}$ and $\mathbf{LR}_{Ex}$ after empirically we found the models did not reach full convergence results with the same learning rates.

Computation The experiments were conducted on machines equipped with Nvidia GPUs, including A5000, A6000, V-100, and A-100 models.

For example, in a LoRA fine-tuning setup with $10,000$ training steps, a batch size of 8 sentences, and a truncation cutoff at 256 tokens, the training process takes approximately 90 minutes for Llama-3.2-3B on an A5000 24GB GPU, for both MAS and causal attention. For Llama-3.2-1B, the same process takes around 40 minutes. As we demonstrated, this duration is sufficient to adapt an existing GPT model to work effectively with MAS.

B Trained Matrices Ablation Study

In section 5, we focused on fine-tuning only two attention matrices ,W_q, W_v. The original implementation of our experiments by Liu et al. (2024) fine-tuned five matrices: the attention matrices W_q, W_k, W_v and the MLP's W_D, W_U. For completeness, we provide full results for the setup of fine-tuning the five matrices.

The results in Table 8 reveal a pattern similar to fine-tuning only the W_q and W_v matrices, showing robust and consistent improvements in average performance across all models. However, as with the W_q, W_v setup, MAS does not surpass causal masking on the HellaSwag benchmark. This limitation is likely due to the sequence length cutoff applied during fine-tuning, which constrains MAS's ability to effectively process longer tasks like HellaSwag.

These results further support the idea that MAS is a promising approach for enhancing model performance.

We further explore diverse choices of trained matrices. Using Llama-3.2 we examine additional fine-tuning setups with unique sets of trained models' parameters. Results, summarized in Figure 5, highlight: (i): Training only the W_q, W_v matrices achieves the best performance, with Llama-3.2-3B reaching 82.3% accuracy, compared to 81.7% for W_q, W_k, W_v and 80.4% for W_q, W_k, W_v, W_U, W_D. (ii): Surprisingly, train-

model	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
Llama-3-8B	75.1	88.6	81.0	95.2	87.1	90.7	78.8	84.4	85.1
Llama-3-8B+MAS	76.4	88.9	82.8	94.6	89.3	93.4	84.5	88.6	**87.3**
Llama-3.1-8B	72.4	84.5	79.4	93.1	84.3	86.9	74.1	81.0	82.0
Llama-3.1-8B+MAS	70.6	85.4	80.3	90.6	84.1	88.8	76.5	84.6	**82.6**
Llama-3.2-1B	62.8	74.1	72.8	65.7	67.7	68.6	52.3	65.2	66.2
Llama-3.2-1B+MAS	64.3	77.1	72.5	62.4	71.0	72.9	55.0	68.0	**67.9**
Llama-3.2-3B	70.8	82.5	77.9	89.7	81.1	81.3	67.2	76.6	78.4
Llama-3.2-3B+MAS	70.0	84.4	79.6	87.4	80.4	86.7	72.5	82.0	**80.4**
Mistral-7B-v0.1	62.4	86.3	73.3	93.4	84.4	87.9	76.4	85.0	81.1
Mistral-7B-v0.1+MAS	72.2	86.2	79.0	82.9	85.0	90.1	77.2	83.0	**82.0**
Mistral-7B-v0.3	68.1	85.9	80.2	87.8	83.7	85.3	72.0	84.6	80.8
Mistral-7B-v0.3+MAS	74.1	75.2	80.9	85.0	85.7	87.7	75.4	85.2	**81.1**
Qwen2.5-3B	70.2	85.3	78.7	91.8	80.3	91.9	79.4	84.8	82.8
Qwen2.5-3B+MAS	65.9	86.3	81.4	87.4	83.3	93.8	82.4	89.0	**83.7**
Qwen2.5-7B	74.9	89.9	79.6	95.3	86.0	95.5	88.1	90.6	87.5
Qwen2.5-7B+MAS	76.1	90.7	82.9	94.1	87.5	96.7	89.6	92.6	**88.8**
MAS win rate	62.5	75.0	87.5	0.0	75.0	100.0	100.0	87.5	100.0

Table 8: Results of Commonsense Reasoning Tasks when training the W_q, W_k, W_v matrices and the MLP's matrices W_U, W_D

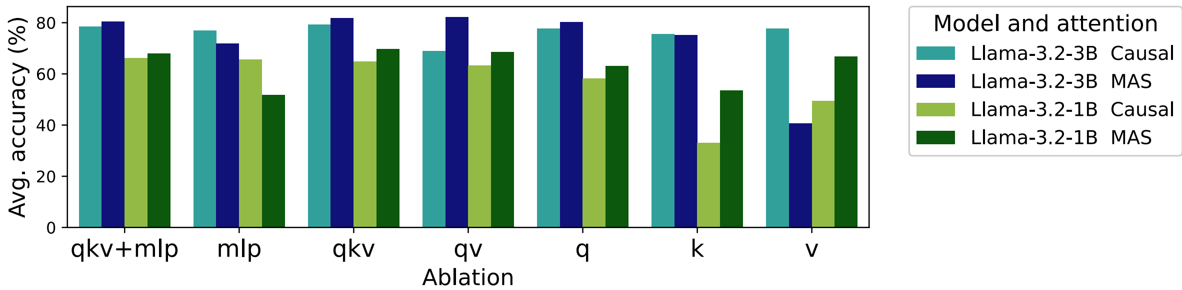

Figure 5: The average accuracy on the Commonsense Reasoning when training different matrices of Llama-3.2.

ing only the W_q matrix achieves 80.3% accuracy, underscoring the critical role of W_q in positional embeddings via the RoPE mechanism. (iii): Across all setups, MAS consistently outperforms causal masking, except when training only the MLP matrices (W_U, W_D). This aligns with the fact that MAS modifies the attention mechanism, making the fine-tuning of the attention matrix essential for leveraging MAS benefits.

C Ablation Study and Training Analysis

This section delves deeper into the fine-tuning process of GPTs using MAS, focusing on the Commonsense Reasoning task with Llama 3.2.

C.1 Hyperparameter Ablation Study

To ensure the robustness of the results presented in section 5, we conducted an ablation study across various hyperparameter configurations. Experiments were repeated with different learning rates and three random seeds. For efficiency, we limited training to 10,000 steps and evaluated on six tasks, excluding HellaSwag and BoolQ due to their higher computational costs.

The results, shown in Figure 6, confirm consistency across seeds, with slight variances for the lowest learning rates. The optimal learning rate of approximately $1e^{-4}$ was consistent with both masking setups, indicating that MAS does not require a different learning rate range. However, MAS demonstrated sensitivity to higher learning rates: beyond a certain threshold, performance degradation was more pronounced with MAS compared to causal masking. This suggests that MAS requires more careful learning rate tuning to achieve optimal results.

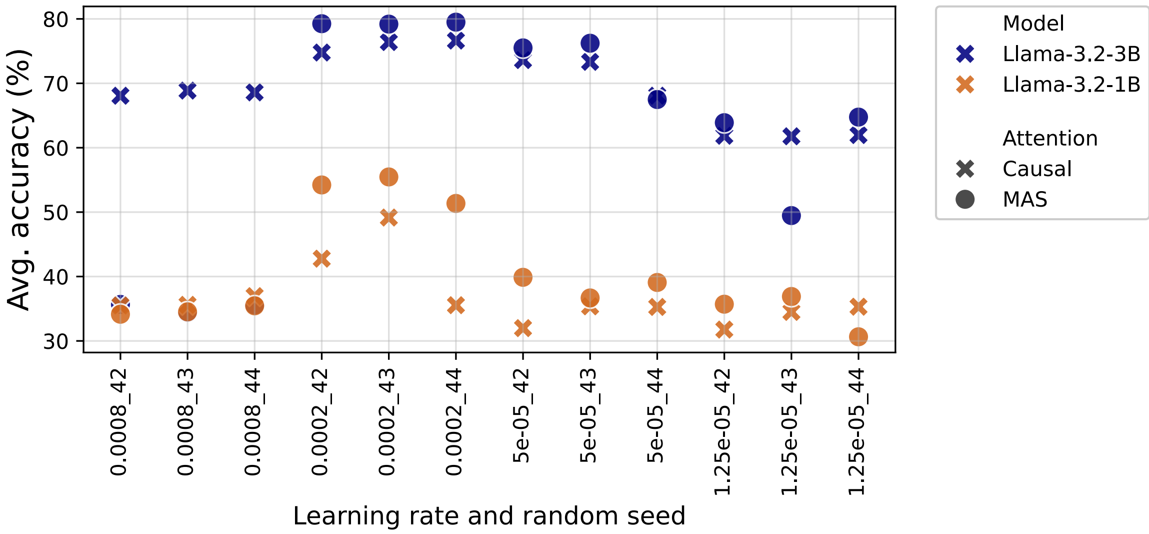

Figure 6: Average accuracy on 6 of the Commonsense Reasoning tasks, as a function of random seeds and learning rates.

C.2 System and Users Prompts Segments

In section 4, we discussed our decision to separate the system prompt and the user prompt into two segments for computational efficiency. Here, we investigate whether this separation also affects the model's performance.

In Figure 7, we present the results of MAS's performance when the system and user prompts are combined into a single segment. The average accuracy across the Commonsense Reasoning benchmark show minimal differences compared to when the prompts are separated, suggesting that the separation does not significantly impact MAS's performance.

We hypothesize that the system prompt, being consistent across tasks, provides limited additional information for the model. Instead, the variability in the user prompt containing the actual question — is the primary driver of the model's performance.

Thus, separating the system and user prompts is justified not only by computational efficiency but also by empirical evidence, supporting its use as a practical and effective approach.

D MAS visualization

In Figure 5, we visualized attention maps for models fine-tuned with MAS. Here, Figure 8 provides additional visualizations of two Llama-3.2-1B instances, one fine-tuned with standard causal masking and the other with MAS.

Identifying the specific function performed by each attention head remains an active area of research (Zheng et al., 2024), which lies outside the scope of this work. The visualizations are presented with two primary objectives: (i) To illustrate how MAS modifies attention patterns compared to standard causal masking. (ii) To provide a resource for future research aimed at gaining deeper insights into the behavior of models fine-tuned with MAS, as well as those trained using causal masking.

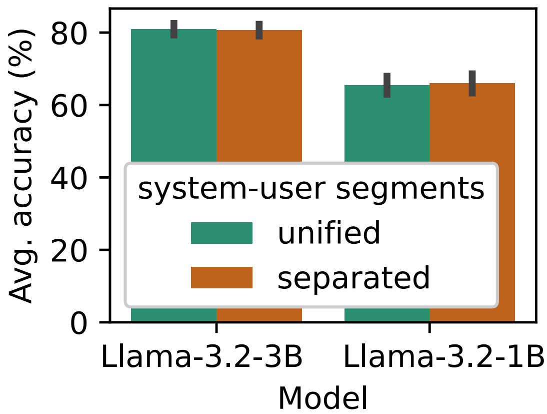

Figure 7: Average accuracy on the Commonsense Reasoning tasks for model with MAS, once where the system prompts and the user prompts are separated into two segments, and once when they are unified into one segment.

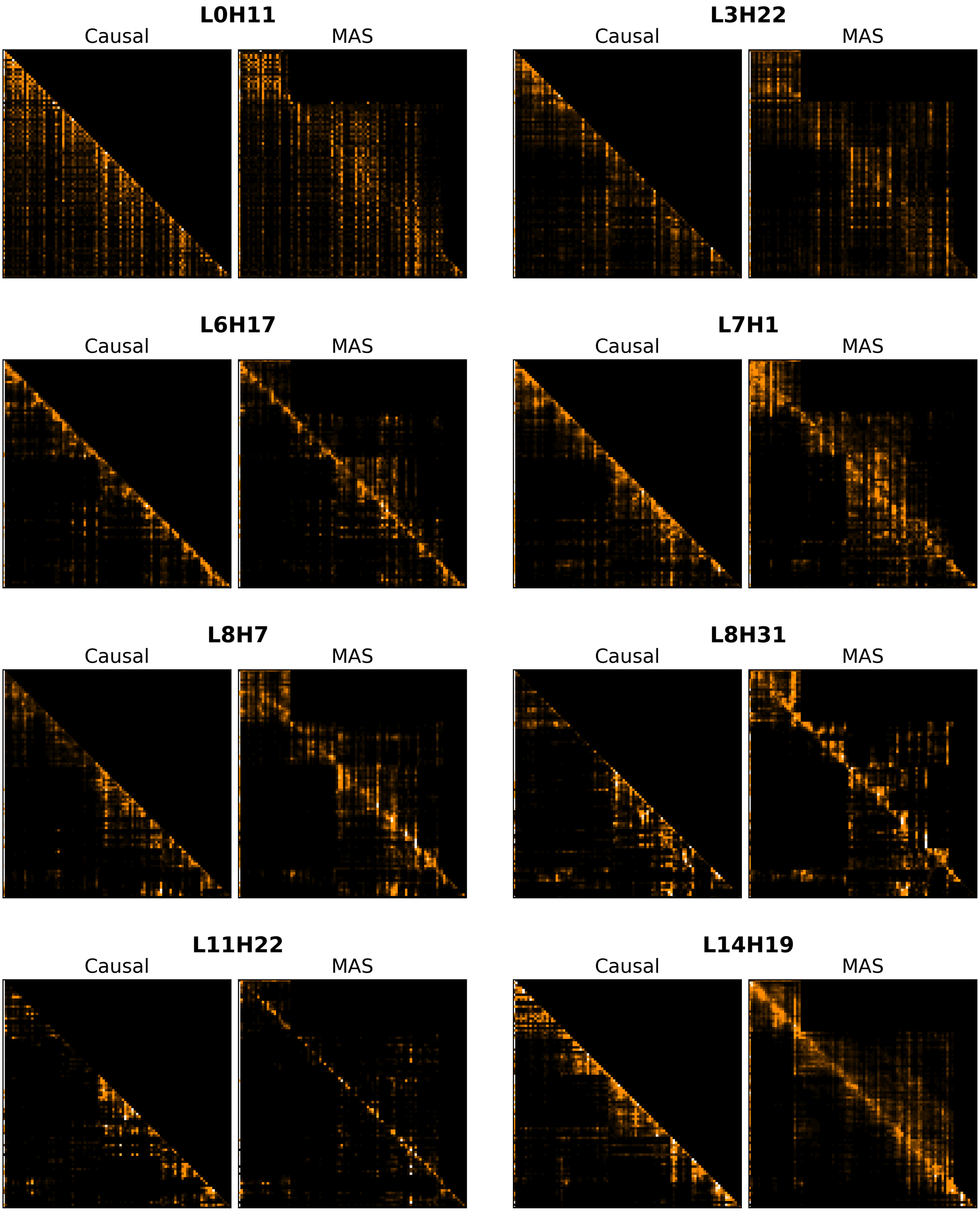

Figure 8: The attention maps of two instances of fine-tuned Llama-3.2-1B, one with standard causal masking and the other with MAS