Mixture of In-Context Experts Enhance LLMs' Long Context Awareness

Hongzhan Lin[1*] **Ang Lv**[1*] **Yuhan Chen**[2*]
Chen Zhu[3] **Yang Song**[4†] **Hengshu Zhu**[3] **Rui Yan**[1†]
[1] Gaoling School of Artificial Intelligence, Renmin University of China
[2] XiaoMi AI Lab [3] Career Science Lab, BOSS Zhipin [4] NLP Center, BOSS Zhipin
{linhongzhan, anglv, ruiyan}@ruc.edu.cn
{chenyuhan5}@xiaomi.com

Abstract

Many studies have revealed that large language models (LLMs) exhibit uneven awareness of different contextual positions. Their limited context awareness can lead to overlooking critical information and subsequent task failures. While several approaches have been proposed to enhance LLMs' context awareness, achieving both effectiveness and efficiency remains challenging. In this paper, for LLMs utilizing RoPE as position embeddings, we introduce a novel method called "Mixture of In-Context Experts" (MoICE) to address this challenge. MoICE comprises two key components: a router integrated into each attention head within LLMs and a lightweight router-only training optimization strategy: (1) MoICE views each RoPE angle as an 'in-context' expert, demonstrated to be capable of directing the attention of a head to specific contextual positions. Consequently, each attention head flexibly processes tokens using multiple RoPE angles dynamically selected by the router to attend to the needed positions. This approach mitigates the risk of overlooking essential contextual information. (2) The router-only training strategy entails freezing LLM parameters and exclusively updating routers for only a few steps. When applied to open-source LLMs including Llama, Mistral and Qwen, MoICE surpasses prior methods across multiple tasks on long context understanding and generation, all while maintaining commendable inference efficiency. Moreover, we also demonstrate the effectiveness of MoICE in pre-training a language model from scratch.

1 Introduction

Although large language models (LLMs) have demonstrated impressive capabilities across diverse NLP tasks, several studies [27, 8, 31] have pointed out that the contextual awareness of LLMs is not as powerful as widely believed, constraining their application in tasks demanding extensive contextual awareness, such as in-context learning [28, 46], coherent long text generation [49, 26] and Retrieval-Augmented Generation (RAG, [19, 6, 10]) tasks necessitating in-context retrieval [8]. Liu et al. [27] identified a common issue termed the "lost-in-middle" phenomenon, indicating that LLMs often exhibit a weaker awareness of information situated in the middle of the long context compared to the beginning or end. Chen et al. [8] highlighted challenges arising from a mathematical property of RoPE [38], a wide-used positional embedding in LLMs, which impedes attention to

[*]Equal contribution. Hongzhan Lin and Ang Lv proposed the idea of MoICE. Hongzhan Lin and Yuhan Chen designed the MoICE router architecture and implemented efficient code. Experiments were conducted by Hongzhan Lin, while Ang Lv led the writing. Code is available at https://github.com/p1nksnow/MoICE.

[†]Corresponding authors: Rui Yan (ruiyan@ruc.edu.cn) and Yang Song (songyang@kanzhun.com)

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

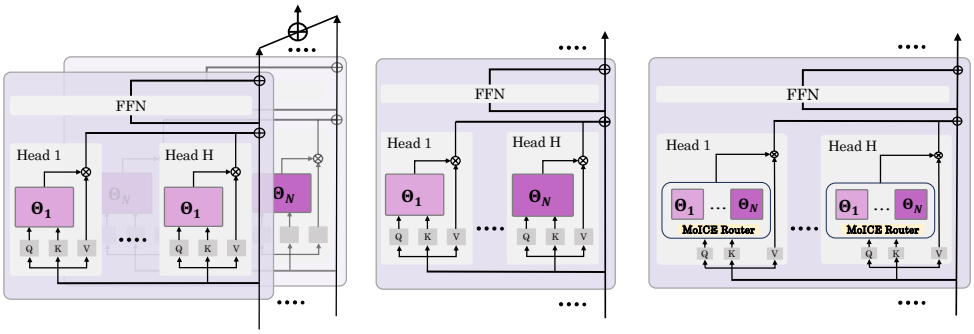

Figure 1: Some methods developed to enhance LLMs' context awareness. (a) Attention Buckets [8] selects N different RoPEs and conducts N parallel inferences for each input. The outputs are then aggregated in the final layer. (b) Ms-PoE [49] employs a unique RoPE angle for each attention head. However, it needs an additional forward pass for RoPE angle assignment. (c) MoICE integrates a router within each attention head. This novel plug-in selects several of the most suitable RoPE angles for each token. The selected RoPE angles collectively contribute to computing the attention scores. MoICE demonstrates superior memory efficiency and performance.

specific positions within the long context. Consequently, if critical information coincides with such positions, task performance suffers.

Many works [23, 49, 8, 48] have attempted to enhance the long-context awareness of LLMs. Central to these efforts is the enhancement of attention heads which serve as the linchpin for contextual awareness, given that FFNs in language models do not introduce token interaction. Chen et al. [8] proposed an inference algorithm named *Attention Buckets* (AB), which enhanced the context awareness of LLMs by executing N inference instances, each with a distinct RoPE angle, and aggregated the outputs at the final layer. Zhang et al. [49] observed the varying awareness of attention heads to contextual positions. They proposed an inference algorithm named *Ms-PoE*. Ms-PoE enhances the utility of position-aware heads by re-scaling the positional embedding indices, equivalent to assigning each head a unique RoPE angle. Figure 1 illustrates these approaches. However, these approaches each come with their own drawbacks: AB conducts excessive redundant FFNs calculations, leading to high memory consumption. In Ms-PoE, determining a distinct re-scale factor for every attention head needs an additional forward pass. Meanwhile, each attention head still depends on a single re-scaled static RoPE. As highlighted by AB [8], this leads to limited awareness of certain contextual positions, thereby constraining its potential. Moreover, a significant drawback of both AB and Ms-PoE lies in their static assignment of the RoPE angle for each attention head throughout the generation. However, as the generation progresses, the positions of crucial tokens shift, necessitating corresponding adjustments in the required RoPE angles for each head.

In this study, we present *Mixture of In-Context Experts* (MoICE), a novel plug-in of LLMs for enhancing context awareness. Specifically, We conceptualize a unique RoPE angle as an "in-context expert," as it can allocate a head's more attention to certain contextual positions [8]. We integrate a router within each attention head, which discerns the potentially important tokens for the head and dynamically selects K RoPE angles that provide comprehensive awareness of these tokens for attention computation. Through the re-computation of only a few query-key dot products, attention patterns computed with selected RoPE angles are aggregated to produce the final attention pattern. This approach yields two primary advantages: (1) It eliminates unnecessary computational overhead in AB, enhancing efficiency. (2) The dynamic expert selection of each head for arbitrary tokens introduces flexibility not attained in previous studies. This minimizes the risk of the initial RoPE angle assigned to a head failing to work due to crucial token positions shifting during generation.

Consequently, MoICE not only surpasses AB's effectiveness but also achieves commendable efficiency. We name our approach as "Mixture of In-Context Experts" (MoICE) due to the aggregation of attention patterns calculated with different RoPE angles resembling the concept of "Mixture of Experts" (MoE, [37]). When applying MoICE to open-source LLMs, we freeze LLMs' parameters

and conduct lightweight training only on the MoICE routers. With only a few quick updates, MoICE surpasses many competitive baselines in tasks involving long-context generation and understanding.

In summary, our main contribution is the introduction of MoICE, a novel plug-in for enhancing LLMs' context awareness. It achieves head-and token-specific dynamic multiple RoPE angles assignment, outperforms previous methods across various tasks, and maintains commendable inference efficiency.

2 Background

We introduce some background of *Mixture of In-Context Experts*, including (1) the rotary position embeddings commonly used by mainstream LLMs, (2) the primary problem addressed in this paper: the limited context awareness of LLMs, (3) an explanation of the underlying reasons for this limitation.

Position embedding Positional embedding is crucial for Transformer [43] to perceive sequence order and compensate for the position-agnostic nature of the attention mechanism. In this paper, we mainly focus on LLMs using Rotary Position Embedding (RoPE, [38]) which is the prevalent position embedding in current LLMs. We discuss other position embeddings in Appendix D.

In a Transformer layer with H attention heads employing RoPE, where d represents the hidden state dimension of each attention head, let $\mathbf{q}_n^h$ and $\mathbf{k}_m^h$ denote the query vector at position n and key vector at position m in the h-th head. To encode position information, RoPE initially applies a rotary matrix to the query and key vectors:

$$\hat{\mathbf{q}}_n^h = \mathbf{R}_{\Theta_j,n} \cdot \mathbf{q}_n \in \mathbb{R}^d, \quad \hat{\mathbf{k}}_m^h = \mathbf{R}_{\Theta_j,m} \cdot \mathbf{k}_m \in \mathbb{R}^d, \tag{1}$$

$$\mathbf{R}_{\Theta_j,n} = \begin{bmatrix} \mathbf{r}_{\theta_{j,0},n} & O & \cdots & O \\ O & \mathbf{r}_{\theta_{j,1},n} & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & \mathbf{r}_{\theta_{j,d/2-1},n} \end{bmatrix}, \text{ where } \mathbf{r}_{\theta_{j,i},n} = \begin{bmatrix} \cos n\theta_{j,i} & -\sin n\theta_{j,i} \\ \sin n\theta_{j,i} & \cos n\theta_{j,i} \end{bmatrix}, O = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{2}$$

Here, $\theta_{j,i} = B_j^{-2i/d}, i \in [0, \cdots, d/2-1]$, is termed as the rotary angle of RoPE, and B_j is typically a fixed base. The subscript j serves to differentiate various RoPE angles Θ, each associated with a distinct B_j, a distinction necessary for discussions in Section 3. This approach effectively incorporates relative position information between m and n in the query-key product during attention computation:

$$\hat{\mathbf{q}}_n^{h\top} \cdot \hat{\mathbf{k}}_m^h = \left(\mathbf{R}_{\Theta_j,n} \cdot \mathbf{q}_n^h\right)^\top \left(\mathbf{R}_{\Theta_j,m} \cdot \mathbf{k}_m^h\right) = \mathbf{q}_n^{h\top} \cdot \mathbf{R}_{\Theta_j,m-n} \cdot \mathbf{k}_m^h, \tag{3}$$

$$\mathbf{Attn}_{nm}^h = \texttt{Softmax}\left(\frac{\hat{\mathbf{q}}_n^{h\top} \cdot \hat{\mathbf{k}}_m^h}{\sqrt{d}}\right) = \texttt{Softmax}\left(\frac{\mathbf{q}_n^{h\top} \cdot \mathbf{R}_{\Theta_j,m-n} \cdot \mathbf{k}_m^h}{\sqrt{d}}\right). \tag{4}$$

Here, $\mathbf{Attn}_{nm}^h$ denotes the attention score assigned by the h-th head at position n to position m.

Context awareness of LLMs LLMs struggle with limited context awareness, significantly impacting their performance in tasks like long-text generation [49], Retrieval-Augmented Generation (RAG, [19, 6, 10, 40]), and multi-turn human-agent interactions [8] involving complex contexts. Liu et al. [27] identified a problem known as "Lost-in-the-Middle," where LLMs process the beginning and end of the context well but have reduced awareness of the middle. Chen et al. [8] observed that LLMs using RoPE exhibit uneven context awareness, favoring certain positions. Peysakhovich et al. [34] further highlighted that LLMs exhibit variable attention to document-level token segments based on their contextual positions. Lv et al. [29] observed that language models develop context awareness, especially in their ability to copy, through "grokking [35]." They suggest pre-training models with increased regularization to enhance this capability.

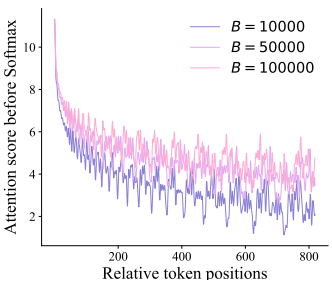

Figure 2: Different Θ_j alter the upper bounds of attention scores between a token and its x-distance neighbors. Each angle is distinguished by its own base value B_j.

Attention waveforms According to Chen et al. [8], LLM's uneven awareness of different contextual positions is due to RoPE's mathematical characteristics. Within RoPE, the attention score exhibits "waveforms" when retrieving the same token from the context, based on their relative positions. The troughs in these waveforms can impair task performance, especially when critical tokens are situated at these positions during generation. Different RoPE angles produce waveforms with troughs occurring at different positions. These phenomena are depicted in Figure 2. A detailed derivation of the depicted curves in Figure 2 is provided in Appendix B.

3 Mixture of In-Context Experts

In this section, we first introduce the core component of MoICE, the MoICE router, detailed in Section 3.1. Subsequently, we delve into the optimization of MoICE in Section 3.2. Figure 3 provides an overview of MoICE. The discussion in this section focuses solely on a single layer of transformer for clarity, with the same principles applying to any other layers.

3.1 Architecture

We aim to design an enhanced attention mechanism in LLMs that dynamically attends to crucial information across various contextual positions required for completing the head's function. As a result, we can mitigate the performance drop caused by inadequate context awareness. Motivated by insights of Chen et al. [8], who demonstrated that a distinct RoPE angle Θ_j could direct the attention heads more focus on specific contextual positions, we propose the integration of a contextual-aware routing mechanism. This routing mechanism is designed to select the appropriate RoPE angles for processing a token. We implement the router as a Multi-Layer Perceptron (MLP) with the `SiLU` activation function:

$$\texttt{Router}\left(\mathbf{q}\right) := \mathbf{W}_3\left(\texttt{SiLU}\left(\mathbf{W}_1\mathbf{q}\right) \odot \left(\mathbf{W}_2\mathbf{q}\right)\right). \tag{5}$$

Here, $\mathbf{q}$ is the query vector that encapsulates the contextual information for the task. This router input indicates the specific information for which the current token is "querying." $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{N\times d}$, and $\mathbf{W}_3 \in \mathbb{R}^{N\times N}$ are weight matrices, where N denotes the number of the number of RoPE angle candidates. Considering each head's distinct function [32, 45, 30], we integrate a router into every attention head in the LLM. Notably, a router's decision is independent of other heads and dynamic to the context.

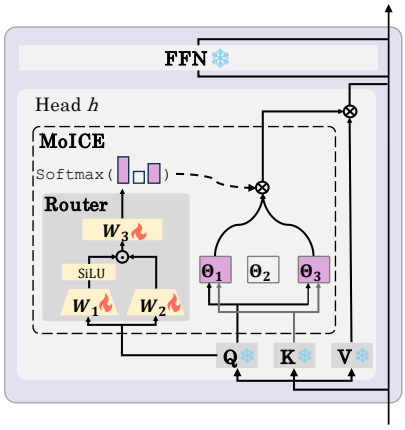

Figure 3: The structure of MoICE. Only the router's parameters are trainable when plugged into an LLM. For clarity, the figure illustrates a single head, with N=3 and K=2 as toy demonstration examples.

As defined in Eq. 5, the router outputs an N-tuple distribution, indicating the weight it allocates for each RoPE angle. In each step in generation, the router selects K angles from a set of N angles $\{\Theta_1, \Theta_2, \ldots\ldots, \Theta_N\}$ for attention computation. Specifically, we first identify the K RoPE angles with the highest routing weights and normalize their relative weights using the `Softmax` function, resulting in $\mathbf{p}_n^h \in \mathbb{R}^K$, representing the probability distribution over the selected RoPE angles within the h-th head:

$$\begin{aligned} \text{TopK-Indices}_n^h &= \texttt{argsort}(\texttt{Router}\left(\mathbf{q}_n^h\right))[:K], \\ \mathbf{p}_n^h &= \texttt{Softmax}\left(\texttt{Router}\left(\mathbf{q}_n^h\right)[\text{TopK-Indices}_n^h]\right), \end{aligned} \tag{6}$$

where $\mathbf{q}_n^h$ represents the query at position n within the h-th attention head in a Transformer layer. Subsequently, we aggregate the attention scores computed with these chosen K in-context experts based on their routing weights to derive the final attention scores for head h from position n to position m:

$$\mathbf{Attn}_{nm}^h = \sum_{j\in\text{TopK-Indices}_n^h} \mathbf{p}_n^h[j] \cdot \texttt{Softmax}\left(\frac{\mathbf{q}_n^{h\top}\cdot \mathbf{R}_{\Theta_j, m-n}\cdot \mathbf{k}_m^h}{\sqrt{d}}\right). \tag{7}$$

Considering RoPE angles impact how attention heads allocate attention and focus on specific contextual positions, we view each distinct RoPE angle as an in-context expert, in contrast to traditional in-weight experts [16, 21, 12, 18], where the experts are learnable parameter weights. Given that these in-context experts together augment LLMs' context awareness, we term this method Mixture of In-Context Experts (MoICE, Section 3). Figure 3 illustrates the overview of MoICE. Our proposed MoICE has three major advantages:

(1) We only add additional computational overhead to the query-key dot products, resulting in a minimal increase in memory usage and a negligible impact on inference speed (Section 4.2).

(2) MoICE dynamically selects suitable RoPE angles token-wise and head-wise, offering unprecedented flexibility and unlocking the full potential of each attention head.

(3) Concerning LLMs' context awareness enhancement, MoICE addresses a longstanding issue: the relative position of the relevant information will shift during generation, leading to previous static modification of the attention heads [8, 47] will be sub-optimal during practical generation. The contextual-aware dynamic routing in MoICE is not bothered by this issue.

3.2 Router-only training

To train the newly incorporated MoICE routers in LLMs, the most straightforward way is to simultaneously update the LLMs' parameters alongside the routers. However, updating the original LLMs' parameters can result in catastrophic forgetting. Therefore, we propose a more effective and efficient strategy, the router-only training strategy, which freezes the LLMs' parameters and solely optimizing the routers.

Given an input sequence, we calculate the negative log-likelihood loss ($\mathcal{L}_{nll}$) for language modeling. During backward propagation, only the parameters of MoICE routers are updated. To mitigate the possibility of a router favoring specific experts disproportionately [16, 44], we incorporate an auxiliary loss $\mathcal{L}_{aux}$ following [16]. An ablation study on this auxiliary loss is in Table 11. Given an input of T tokens and N experts, we calculate the $\mathcal{L}_{lb}$ by the scaled dot-product between frequency vector $\mathbf{F}$ and probability vector $\mathbf{P}$:

$$\mathcal{L}_{aux} = \alpha \cdot N \cdot \sum_{j=1}^{N} \mathbf{F}_j \cdot \mathbf{P}_j. \tag{8}$$

Eq. 8 avoids the router falling into a sub-optimal solution favoring specific experts overwhelmingly, as its minimal is achieved when the routing probability is uniform. Here, α is the weighting factor for load balancing loss. $\mathbf{F}_j$ denotes the proportion of the j-th expert selected across all positions and attention heads, while $\mathbf{P}_j$ denotes the proportion of router weight assigned to expert j:

$$\begin{aligned} \mathbf{F}_j &= \frac{1}{T \times H} \sum_{t=1}^{T} \sum_{h=1}^{H} \mathbb{1}\{j \in \text{TopK-Indices}_t^h\}, \\ \mathbf{P}_j &= \frac{1}{T \times H} \sum_{t=1}^{T} \sum_{h=1}^{H} \mathbb{1}\{j \in \text{TopK-Indices}_t^h\} \cdot \mathbf{p}_t^h[j]. \end{aligned} \tag{9}$$

Our overall training objective is to minimize the following loss:

$$\mathcal{L} = \mathcal{L}_{nll} + \mathcal{L}_{aux}. \tag{10}$$

4 Experiment

4.1 Setup

To evaluate the efficacy of MoICE, we implement it with open-source LLMs, which we will introduce later, and conduct lightweight training of MoICE routers on a small and general dataset. Subsequently, we evaluate the enhanced LLM's capability to zero-shot undertake multiple tasks in long context understanding and generation, as detailed in Section 4.2 and Section 4.3.

Training data We use a training dataset[3] which extracts the one thousand longest entries from OpenHermes [41]. OpenHermes is a multi-source integrated dataset containing high-quality synthetically generated instruction and chat samples. A detailed analysis of other training data is in Section 5.3.

Hyperparameters for MoICE-router-only training We froze all the original parameters of the open-source LLMs we used and only trained the MoICE router. Following Attention Buckets [8], we employed the RoPE angle set of $N = 7$ items, each assigned with base values as follows: $\{1.0 \times 10^4, 1.75 \times 10^4, 1.8 \times 10^4, 1.9 \times 10^4, 2.0 \times 10^4, 2.25 \times 10^4, 2.5 \times 10^4\}$ for Llama2-7B and Mistral-7B, $\{1.0 \times 10^6, 1.25 \times 10^6, 1.4 \times 10^6, 1.8 \times 10^6, 1.9 \times 10^6, 2.25 \times 10^6, 2.5 \times 10^6\}$ for Qwen1.5-7B. By default, unless otherwise specified, the attention head selects K=7 bases to ensure a fair comparison with [8]. Section 4.2 introduces our baselines in detail. Section 5 delves into the impact of set size and the number of selected items.

We implement a warm-up strategy comprising 20% of the total steps, with a maximum learning rate of 0.0001. The batch size is 128. α is set as 0.3. We train the MoICE routers for 1 epoch (about 8 minutes) on four A800-80G GPUs.

4.2 Long context understanding and generation

Following the L-Eval benchmark [1], we evaluated the LLM with tasks categorized into two main groups: closed-ended and open-ended tasks. Closed-ended tasks primarily focus on the capacity for understanding and reasoning within long contexts, including tasks like multiple-choice questions from QuALITY [5], Coursera, [4] TOEFL [11], and True/False question answering from SFiction. [5] On the other hand, open-ended tasks include summarization generation and open-format question-answering tasks, requiring extracting information from lengthy in-context documents. The open-ended tasks comprise a subset of 181 questions drawn from 29 diverse long documents.

Baselines and open-source LLMs In evaluating the efficacy of our proposed MoICE, we compare it against several state-of-the-art methods known for enhancing the capacity of LLMs to understand and generate long contexts. These baselines include two context extrapolation techniques: Positional Interpolation (PI, [7]) and Dynamic NTK [15]. Additionally, we consider two inference algorithms for context-awareness enhancement: Ms-PoE [49] and Attention Buckets [8].

We evaluate all these methods alongside our MoICE on three representative open-source LLMs that utilize RoPE for positional embeddings: Llama2-7B-Chat [42], Mistral-7B-Instruct-v0.1 [20] and Qwen1.5-7B-Chat [2]. Llama2-7B and Qwen1.5-7B support a pre-trained context length of 4,096 and 32,768, respectively. Mistral-7B employs a sliding window attention (SWA) mechanism with a window size of 4,096 tokens, enabling it to accommodate longer contexts than the default. Therefore, we conduct experiments with a context length of 8,192 on Mistral-7B, using SWA as the exclusive baseline for comparison. For PI and Dynamic NTK, we apply a scaling ratio of 1.5, while for the remaining baselines, we adhere to the hyperparameters specified in their original papers. All methods are tested on a single A800-80G GPU, except for applying AB to Mistral-7B-8k, which needs 2 GPUs due to substantial memory requirements.

Evaluation metrics We adopt the exact match for closed-ended tasks. For open-ended tasks, we employ *GPT-4-Turbo* [33] as the judge to evaluate the effectiveness of various enhancement methods on open-source LLMs. This evaluation compares their performance against *GPT3.5-Turbo-16k-0613* across 181 questions.

Results and analysis We report our experimental results in Table 1. MoICE significantly enhances the overall performance of Llama-2-7B-chat (with p-value < 0.02 in the t-test) in both closed-ended and open-ended tasks. On Mistral, MoICE outperforms all baseline models significantly (p-value < 0.02). We also report the mean and standard deviation of MoICE in Table 10. Standard fine-tuning degrades the performance of original LLMs, demonstrating catastrophic forgetting and proving that the improvement of MoICE does not stem from more training. These results underscore MoICE's

[3] `https://huggingface.co/datasets/HuggingFaceH4/OpenHermes-2.5-1k-longest`

[4] `https://coursera.org/`

[5] `https://github.com/nschaetti/SFGram-dataset`

Table 1: Experimental results on the L-Eval Benchmark [1]. Applying to various models, MoICE demonstrate superior performance compared to previous competitive approaches. We emphasize the highest score in bold.

Method	Closed - Ended Task					Open - Ended Task		
	Coursera	QuALITY	TOEFL	SFiction	Average	wins	ties	win-rate%*
Llama2-7B-Chat [42]	36.77	38.12	55.02	60.16	47.52	68	117	34.94
+ Fine-tuning	32.85	30.20	51.30	59.38	43.43	65	91	30.52
+ PI [7]	38.23	38.61	56.51	61.72	48.77	76	112	36.46
+ Dynamic NTK [15]	40.26	39.11	55.76	62.50	49.41	82	112	38.12
+ Ms-PoE [49]	39.24	40.10	55.76	63.28	49.60	86	110	38.95
+ AB [8]	**40.41**	41.09	**56.88**	61.72	50.02	85	114	39.23
+ MoICE (Ours)	39.83	**42.08**	56.13	**64.84**	**50.72**	**89**	**118**	**40.88**
Mistral-7B-Instruct-8k [20]	45.20	44.06	62.08	61.72	53.27	71	105	34.11
+ Fine-tuning	25.29	26.73	25.65	50.00	31.92	53	85	26.38
+ SWA	44.77	42.57	62.08	60.94	52.59	73	89	32.45
+ PI [7]	44.19	44.06	**64.68**	62.50	53.86	73	96	33.43
+ Dynamic NTK [15]	45.35	42.08	62.08	**63.28**	53.20	78	103	35.77
+ Ms-PoE [49]	46.37	45.05	61.34	57.03	52.45	84	106	37.84
+ AB [8]	46.08	42.57	62.08	62.50	53.31	**87**	110	39.22
+ MoICE (Ours)	**47.82**	**46.53**	**64.68**	62.50	**55.38**	85	**117**	**39.36**
Qwen1.5-7B-Chat [2]	**78.44**	61.88	61.19	69.53	67.76	83	**119**	40.83
+ PI [7]	76.58	61.88	60.32	70.31	67.27	83	107	39.11
+ Dynamic NTK [15]	78.07	62.38	60.32	70.31	67.77	84	111	40.20
+ Ms-PoE [49]	75.47	60.89	60.47	**71.88**	67.18	OOM	OOM	N/A
+ AB [8]	**78.44**	OOM	OOM	OOM	N/A	OOM	OOM	N/A
+ MoICE (Ours)	**78.44**	**62.87**	**61.77**	71.09	**68.54**	**91**	105	**41.59**

* Following [1], win-rate = (win counts + 0.5 * tie counts)

Table 2: Practical inference time (in minutes) / GPU memory costs (GB) on a single A800-80G GPU for each method applied to Llama2-7B-Chat (top) and Mistral-7B-Instruct-8k (bottom), respectively. Due to out-of-memory issues, AB can not accomplish many tasks, denoted as OOM in the table.

Method	Coursera ↓	QuALITY ↓	TOEFL ↓	SFiction ↓	Open-Ended ↓	Average ↓
AB [8]	10.9 / 78.7	18.1 / 62.5	19.9 / 56.5	5.0 / 33.2	45.9 / 78.2	20.0 / 61.8
Ms-PoE [49]	4.1 / 27.2	6.0 / 27.8	6.7 / 28.6	6.0 / 27.8	20.2 / 28.9	8.6 / 28.1
MoICE (Ours)	5.0 / 19.6	11.0 / 19.7	10.2 / 19.5	1.6 / 15.2	34.2 / 23.2	12.4 / 19.4
AB [8]	OOM	OOM	37.2 / 71.4	OOM	OOM	N/A
Ms-PoE [49]	14.1 / 50.3	11.2 / 48.4	9.8 / 25.4	4.5 / 50.3	72.8 / 62.4	22.5 / 47.4
MoICE (Ours)	13.4 / 25.7	7.7 / 22.9	11.3 / 20.4	2.3 / 22.8	77.8 / 29.3	22.5 / 24.2

efficacy in enhancing LLMs' ability to understand and generate long contexts, both of which require high context awareness. Furthermore, these results underscore the broad applicability of MoICE across different LLMs.

Regarding efficiency, we provide practical inference time and memory costs associated with AB, Ms-PoE, and MoICE in Table 2. For a fair comparison, we utilize Flash Attention 2 [13] across all approaches. While achieving superior overall performance, MoICE remains at an inference speed similar to Ms-PoE and notably excels in memory efficiency compared to these two baselines.

We also perform further experiments on one additional long context benchmark LongBench [4], which are detailed in Appendix A.

4.3 Retrieval-augmented generation (RAG)

Retrieval-augmented generation (RAG) tasks involve retrieving numerous documents related to the current generation. The retrieved documents are arranged in the context. RAG necessitates that LLMs have robust context awareness to pinpoint crucial documents, process the retrieved information effectively, and integrate it to generate responses.

Following [8, 49], we employ the MDQA task to evaluate the efficacy of MoICE in enhancing LLMs' performance in RAG tasks. Meanwhile, MDQA offers the bonus of allowing flexible control over

Table 3: The experiment results on the MDQA task. MoICE achieve superior average performance compared to previous competitive approaches. We emphasize the highest score in bold.

Method	**1**	**3**	**5**	**7**	**10**	**Gap**	**Avg.**
Llama2-7B-Chat	64.14	65.95	64.97	62.67	**67.53**	4.86	65.05
+ Ms-PoE [49]	66.06	64.29	63.99	62.22	64.75	3.84	64.34
+ AB [7]	**66.36**	66.14	65.25	63.20	64.93	3.16	65.18
+ MoICE (Ours)	65.50	**66.33**	**65.61**	**64.11**	65.84	**2.22**	**65.48**
Method	**1**	**8**	**15**	**23**	**30**	**Gap**	**Avg.**
Mistral-7B-Instruct-8k	58.38	47.42	46.97	49.68	50.81	**11.41**	50.65
+ Ms-PoE [49]	52.76	41.24	42.80	42.90	43.58	11.52	44.66
+ AB [7]	58.57	47.57	47.12	49.83	**50.96**	11.45	50.81
+ MoICE (Ours)	**61.81**	**52.54**	**52.43**	**50.36**	49.34	12.47	**53.30**

the location of documents, enabling a more precise evaluation of LLMs' context awareness across various contextual positions.

Our MDQA experiments leverage a subset of NaturalQuestions-Open [25, 24], consisting of 2,655 queries, following [49, 27]. Each query is paired with a context consisting of 10 or 30 documents (with an average of 1,722 or 5,046 tokens), depending on the model (Llama-2-7B-chat or Mistral-7B-Instruct-8k), tasked with answering based on this contextual information. Only one document among these comprises useful information for the given query. We compare Ms-PoE, AB, and MoICE, testing each method through 5 iterations. For Llama, the relevant document is positioned 1st, 3rd, 5th, 7th, and 10th within the context, while for Mistral, it is positioned 1st, 8th, 15th, 23rd, and 30th, respectively.

In Table 3, MoICE on Llama demonstrates the highest average performance across most positions, showcasing its remarkable stability. Its accuracy scores show minimal variation, with only a marginal difference of 2.22 points between its highest and lowest values. On Mistral, MoICE exhibits significant average improvement (p-value < 0.02). Notably, when the relevant document is positioned at the end of the context, all methods on Llama exhibit a decrease compared to the original model, although MoICE shows a minimal decline. This phenomenon also happens in the Mistral model. We posit that this decline may stem from the original model predominantly directing attention towards nearest documents [27, 34]. However, as approaches enhance awareness of various contextual positions, the model's attention to the nearest documents is diffused by other positions, as its overall capacity for context awareness is constant and limited. Nevertheless, MoICE consistently emerges as the superior-performing method overall across language models.

5 Method analysis

In this section, we delve into a comprehensive analysis of the properties of MoICE. We illustrate how N, the total number of in-context experts (Section 5.1), and K, the specific number of selected in-context experts (Section 5.2), influence MoICE. We further demonstrate that MoICE is robust to training data (Section 5.3) Additionally, we present a case study demonstrating the dynamic selection of in-context experts for tokens during generation (Section 5.4). Finally, we verify the effectiveness of language model with MoICE architecture in pretraining stage (Section 5.5).

5.1 The effect of expert total numbers N

We investigate the impact of the total number of experts. Employing the search algorithm proposed by Chen et al. [8], we obtain various sets of different sizes, each comprising complementary base values. The searched expert sets are detailed in Appendix E. We apply MoICE to Llama-2-7B-chat and test the model on L-Eval tasks. The results are presented in Table 4. The results of the original Llama are denoted as (N=1) in the table. As the table illustrates, MoICE demonstrates improvement to LLMs' context awareness with increasing N, with noticeable improvements even when N is as small as 3. However, as N reaches 9, the average performance is close to N=7, indicating a performance plateau. This suggests that having N=7 experts is sufficient for general usage.

Table 4: The performance of Llama-2-7B-chat enhanced by MoICE with N in-context experts. We show results marked with color to emphasize the improvements over the original model.

Method	Coursera	QuALITY	TOEFL	SFiction	Avg.
Original (N=1)	36.77	38.12	55.02	60.16	47.52
N=3	37.65	40.10	55.76	62.50	49.00
N=5	38.23	39.60	**56.13**	63.28	49.32
N=7	39.83	**42.08**	**56.13**	**64.84**	**50.72**
N=9	**40.26**	41.58	**56.13**	**64.84**	50.70

Table 5: The improvement of context awareness of Llama-2-7B-chat by MoICE, wherein each head dynamically selects diverse K experts (N=7). We show results marked with color to emphasize the improvements over the original model.

Method	Coursera	QuALITY	TOEFL	SFiction	Avg.
Original (N=1)	36.77	38.12	55.02	60.16	47.52
K=1	35.03	35.64	**56.51**	61.72	47.22
K=3	**39.83**	41.58	56.13	**64.84**	50.60
K=5	38.52	39.60	56.13	**64.84**	49.77
K=7	**39.83**	**42.08**	56.13	**64.84**	**50.72**
Equal Weights	36.48	38.12	53.90	61.72	47.56
Random Weights	15.55	28.71	21.75	8.59	18.65

5.2 The effect of selected experts number K

With a fixed number of experts (N=7), we examine the effect of different numbers of chosen experts K with values of 1, 3, 5, and 7. We consider two additional setups where 7 experts are selected with equal weights ("Equal Weights") and with random weights ("Random Weights"), using Llama-2-7B-chat as the case study. The results are presented in Table 5. Setting $K = 1$ doesn't enhance or significantly degrade the model's performance, aligning with our assertion in the Introduction (Section 1): assigning a single and unique RoPE angle to each head inadequately explores the head's functionality. For K greater than 3, performance improvements become evident. This shows that the MoICE router in our method can select the appropriate combination of experts to better aware the context. Randomly selecting experts ruins the model's language modeling ability, leading to aberrant outputs.

5.3 MoICE is robust to training data

We further analyze the impact of the data for training routers. We additionally use three instruction fine-tuning datasets from different sources: a self-instruct dataset, Airoboros [22]; and two datasets for LLM alignment with long context, Long-Alpaca [9], and LongAlign [3]. The hyperparameters remain consistent as mentioned in Section 4. As presented in Table 6, MoICE exhibits almost identical scores when trained on different data, showcasing the robustness of our method.

Table 6: The improvement of context awareness of Llama-2-7B-chat by MoICE trained on various data.

Training Data	Coursera	QuALITY	TOEFL	SFiction	Avg.
OpenHermes [41]	39.83	42.08	56.13	64.84	50.72
Airoboros [22]	39.68	41.58	56.13	64.84	50.56
Long-Alpaca [9]	39.68	42.08	56.13	64.84	50.68
LongAlign [3]	39.68	41.58	56.13	64.84	50.56

5.4 The visualization of dynamic routing states

We provide a case study exemplifying the dynamic routing mechanism within MoICE during text generation. Depicted in Figure 4 in the Appendix, the MoICE router of each head independently selects distinct experts. At each step of the generation process, these heads dynamically choose experts for each new token. This dynamic utilization of diverse RoPE angles within each attention

head maximizes the potential of attention heads across various inputs, a capability not attained in prior research, including both Attention Buckets and Ms-PoE.

5.5 Applying MoICE to the pretraining stage

We further evaluate the performance of a language model with MoICE architecture in pretraining stage. Specifically, we train a language model with a Llama architecture of 49M parameters, with and without MoICE respectively. We pretrain a small model from scratch and observe the effectiveness of MoICE. More experimental details can be found in Appendix C. We measure the model's context awareness on the Key-Value Retrieval [27] task, which uses multiple randomly generated key-value string pairs as prompts to evaluate the model's ability to extract the correct value corresponding to a given query from the context. One prompt example can be found in Figure 6.

Table 7: Experimental results on the Key-Value Retrieval task [4], We evaluate the model's awareness of different positions by controlling the position index of the key-value pair corresponding to the query among 10 key-value pairs in the prompt.

Position	**1**	**3**	**5**	**7**	**9**
Baseline	0.476	0.324	0.328	0.344	0.502
+ MoICE	0.652	0.762	0.634	0.622	0.814

From the results in Table 7, we can see that our model can significantly increase the contextual capabilities of the pretrained language model, which indicates the potential of scaling up our method in pretraining stage.

6 Conclusion

In this paper, we introduce a novel approach to enhancing the context awareness of LLMs termed *Mixture of In-Context Experts* (MoICE). Through lightweight training, open-source LLMs such as Llama and Mistral, enhanced by MoICE, demonstrate improved context awareness. Across numerous tasks demanding substantial context awareness, MoICE-enhanced LLMs consistently outperform competitive baselines, all the while maintaining commendable efficiency. A distinctive feature of MoICE is that it first implements head- and token-specific RoPE angles assignment for attention heads, a pivotal factor contributing to its success. This paper underscores the need to address the inherent limitations in current LLMs and advocates for a thorough exploration of their existing capabilities.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (NSFC Grant No. 62122089), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China. Ang Lv is supported by the Outstanding Innovative Talents Cultivation Funded Programs 2024 of Renmin University of China.

References

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

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu,

Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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

[4] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*, 2023.

[5] Samuel R Bowman, Angelica Chen, He He, Nitish Joshi, Johnny Ma, Nikita Nangia, Vishakh Padmakumar, Richard Yuanzhe Pang, Alicia Parrish, Jason Phang, et al. Quality: Question answering with long input texts, yes! *NAACL 2022*, 2022.

[6] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762, 2024.

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

[8] Yuhan Chen, Ang Lv, Ting-En Lin, Changyu Chen, Yuchuan Wu, Fei Huang, Yongbin Li, and Rui Yan. Fortify the shortest stave in attention: Enhancing context awareness of large language models for effective tool use. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11160–11174, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[9] Yukang Chen, Shaozuo Yu, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Long alpaca: Long-context instruction-following models. `https://github.com/dvlab-research/LongLoRA`, 2023.

[10] Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. Lift yourself up: Retrieval-augmented text generation with self-memory. *Advances in Neural Information Processing Systems*, 36, 2024.

[11] Yu-An Chung, Hung-Yi Lee, and James Glass. Supervised and unsupervised transfer learning for question answering. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1585–1594, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[12] Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

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

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[15] emozilla. Dynamically scaled rope further increases performance of long context llama with zero fine-tuning. `https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases`.

[16] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[17] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

[18] Zhuocheng Gong, Ang Lv, Jian Guan, Junxi Yan, Wei Wu, Huishuai Zhang, Minlie Huang, Dongyan Zhao, and Rui Yan. Mixture-of-modules: Reinventing transformers as dynamic assemblies of modules, 2024.

[19] Yizheng Huang and Jimmy Huang. A survey on retrieval-augmented text generation for large language models. *arXiv preprint arXiv:2404.10981*, 2024.

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

[21] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[22] jondurbin. airoboros: using large language models to fine-tune large language models. `https://github.com/jondurbin/airoboros`.

[23] He Junqing, Pan Kunhao, Dong Xiaoqun, Song Zhuoyang, Liu Yibo, Liang Yuxin, Wang Hao, Sun Qianguo, Zhang Songxin, Xie Zejian, et al. Never lost in the middle: Improving large language models via attention strengthening question answering. *arXiv preprint arXiv:2311.09198*, 2023.

[24] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[25] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics.

[26] Jia-Nan Li, Quan Tu, Cunli Mao, Zhengtao Yu, Ji-Rong Wen, and Rui Yan. Streamingdialogue: Prolonged dialogue learning via long context compression with minimal losses. *arXiv preprint arXiv:2403.08312*, 2024.

[27] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.

[28] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[29] Ang Lv, Ruobing Xie, Xingwu Sun, Zhanhui Kang, and Rui Yan. Language models "grok" to copy, 2024.

[30] Ang Lv, Kaiyi Zhang, Yuhan Chen, Yulong Wang, Lifeng Liu, Ji-Rong Wen, Jian Xie, and Rui Yan. Interpreting key mechanisms of factual recall in transformer-based language models, 2024.

[31] Ang Lv, Kaiyi Zhang, Shufang Xie, Quan Tu, Yuhan Chen, Ji-Rong Wen, and Rui Yan. Are we falling in a middle-intelligence trap? an analysis and mitigation of the reversal curse, 2023.

[32] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[33] OpenAI. Gpt-4 technical report, 2024.

[34] Alexander Peysakhovich and Adam Lerer. Attention sorting combats recency bias in long context language models, 2023.

[35] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022.

[36] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022.

[37] Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

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

[39] Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. A length-extrapolatable transformer. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14590–14604, Toronto, Canada, July 2023. Association for Computational Linguistics.

[40] Tao Tan, Yining Qian, Ang Lv, Hongzhan Lin, Songhao Wu, Yongbo Wang, Feng Wang, Jingtong Wu, Xin Lu, and Rui Yan. Pear: Position-embedding-agnostic attention re-weighting enhances retrieval-augmented generation with zero inference overhead, 2024.

[41] Teknium. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants, 2023.

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

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[44] Haotao Wang, Ziyu Jiang, Yuning You, Yan Han, Gaowen Liu, Jayanth Srinivasa, Ramana Kompella, Zhangyang Wang, et al. Graph mixture of experts: Learning on large-scale graphs with explicit diversity modeling. *Advances in Neural Information Processing Systems*, 36, 2024.

[45] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023.

[46] Kaiyi Zhang, Ang Lv, Yuhan Chen, Hansen Ha, Tao Xu, and Rui Yan. Batch-ICL: Effective, efficient, and order-agnostic in-context learning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics ACL 2024*, pages 10728–10739, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics.

[47] Liang Zhang, Katherine Jijo, Spurthi Setty, Eden Chung, Fatima Javid, Natan Vidra, and Tommy Clifford. Enhancing large language model performance to answer questions and extract information more accurately. *arXiv preprint arXiv:2402.01722*, 2024.

[48] Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. Tell your model where to attend: Post-hoc attention steering for llms. *arXiv preprint arXiv:2311.02262*, 2023.

[49] Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, and Zhangyang Wang. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *arXiv preprint arXiv:2403.04797*, 2024.

A Results on LongBench

LongBench [4] is a benchmark for bilingual, multitask, and comprehensive assessment of long context understanding capabilities of large language models. We choose 16 tasks from LongBench, spanning five long-text application scenarios: **Single-Doc QA** (NarrativeQA, Qasper, and MultiFieldQA-en), **Multi-Doc QA** (HotpotQA, 2WikiMQA and Musique), **Summarization** (GovReport, QMSum and MultiNews), **Few-shot Learning** (TREC, TriviaQA, SAMSum and LSHT) and **Synthetic Tasks** (Passage Count, PassageRetrieval-en and PassageRetrieval-zh).

For evaluation, we follow the setup of the LongBench benchmark. The input length of LLama2-7B is set to 4k, Mistral-7B to 8k, and Qwen1.5-7B to 32k. We report the average task scores for each scenario for all methods in Table 8.

Table 8: Experimental results on the LongBench Benchmark [4]. We emphasize the highest score in bold.

Method	Single-Doc QA	Multi-Doc QA	Summarization	Few-shot Learning	Synthetic Tasks	Average
Llama2-7B-Chat [42]	25.54	18.47	23.37	51.78	3.94	24.62
+ PI [7]	23.42	23.73	25.34	51.63	7.63	26.35
+ NTK [15]	24.73	23.67	25.41	51.97	8.33	26.82
+ Ms-PoE [49]	23.68	**24.59**	25.33	51.66	8.04	26.66
+ AB [8]	**27.06**	22.94	25.52	**52.84**	8.62	27.40
+ MoICE (Ours)	26.31	23.70	**25.60**	52.34	**9.71**	**27.53**
Mistral-7B-Instruct-8k [20]	27.20	19.89	24.22	52.41	5.06	25.76
+ PI [7]	30.94	**24.94**	26.24	49.34	**9.35**	28.16
+ NTK [15]	30.46	21.21	23.89	52.41	8.44	27.28
+ Ms-PoE [49]	27.90	17.89	20.28	48.59	8.95	24.72
+ AB [8]	29.81	21.95	25.58	54.42	7.89	27.93
+ MoICE (Ours)	**31.09**	22.98	**26.69**	**55.76**	8.02	**28.91**
Qwen1.5-7B-Chat [2]	34.66	35.91	25.77	56.89	33.83	37.41
+ PI [7]	28.28	17.08	24.60	**57.51**	32.67	32.03
+ NTK [15]	31.35	23.98	24.95	56.64	32.50	33.88
+ Ms-PoE [49]	OOM	OOM	OOM	OOM	OOM	N/A
+ AB [8]	OOM	OOM	OOM	OOM	OOM	N/A
+ MoICE (Ours)	**39.37**	**37.35**	**25.81**	57.29	**34.83**	**38.93**

MoICE consistently shows improved average performance across models with 4k, 8k, and 32k context lengths, surpassing previous competitive approaches.

B Attention waveforms

In this section, we will elaborate on attention waveforms and the concept of complementarity. Assuming $\hat{\mathbf{q}}_n^h \cdot \hat{\mathbf{k}}_m^h$ is the attention score (before softmax) of the n-th position to m-th position on the h-th attention head. The attention score can be formulated as follows:

$$
\begin{aligned}
\hat{\mathbf{q}}_n^h \cdot \hat{\mathbf{k}}_m^h &= \left(\mathbf{R}_{\Theta,n}\mathbf{q}_n\right)^T \left(\mathbf{R}_{\Theta,m}\mathbf{k}_m\right) \\
&= \mathrm{Re}\left[\sum_{j=0}^{d/2-1} \mathbf{q_n^h}[2j:2j+1]\mathbf{k_m^{h*}}[2j:2j+1]e^{i(n-m)\theta_j}\right] \\
&= \sum_{j=0}^{d/2-1}\left(q_{n\,2j}^h \cdot k_{m\,2j}^h + q_{n\,2j+1}^h \cdot k_{m\,2j+1}^h\right)\cos\left((n-m)\theta_j\right) \\
&+ \left(q_{n\,2j}^h \cdot k_{m\,2j+1} - q_{n\,2j+1}^h \cdot k_{m\,2j}^h\right)\sin\left((n-m)\theta_j\right),
\end{aligned}
$$

where $\theta_j = B^{-2j/d}$, B is the rotary base of RoPE. Considering a context-awareness task, basic context awareness relies on attending to the same token and then copying its next token as outputs [32]. To simplify the calculation, we set both $\mathbf{q}_n^h$ and $\mathbf{k}_m^h$ as all-one vectors to observe the impact of relative positions on the attention when retrieving the same token from the context. This impact (or the intensity of attention) is dubbed as attention waveform $\mathcal{W}$ by [8].

$$\mathcal{W} \leq \sum_{j=0}^{d/2-1} 2\cos\left((n-m)\theta_j\right).$$

As illustrated in Figure 2, the waveform exhibits two notable mathematical properties concerning attention scores: it demonstrates fluctuations and undergoes a gradual decay with the increasing relative position (i.e., long-term decay).

Chen et al. [8] observed that crucial information falling within the troughs of a waveform might diminish the performance of models employing RoPE. Meanwhile, they pointed out the waveform, characterized by peaks and troughs, vary across RoPE bases. When leveraging the peaks of one attention wave to compensate for the overlook of the troughs in another, the model's capability to perceive and process information from diverse contextual positions can be enhanced. When a set of bases possesses this waveform characteristic, they are termed "complementary."

C Experimental details on pretraining

In this section, we provide detailed experimental setup in Section 5.5. The model we has 4 layers, 6 heads per layer, a hidden layer dimension of 512 and an intermediate size of 1280. We train the model using the OpenWebText pretraining [17] dataset. We use four GTX A800-80Gs for training for 600k steps, with a context window of 512. During pretraining, we use a learning rate of 0.004 with a cosine annealing schedule and 6,000 warm-up steps.

D Discussions on more position embeddings

In this section, we discuss other position embeddings and demonstrate why they are not studied, e.g., discarded in LLMs, do not exhibit attention waveform pattern, or are in the same family of RoPE: Firstly, the waveform pattern only exists in position embeddings constructed by cosine functions. Regarding the cosine embedding used in the original Transformer, it does exhibit long-term decay and periodic waveforms. However, this embedding is disregarded in modern LLMs. Moreover, these embeddings are incorporated before the initial model layer rather than during the attention computation, making it hard to assess their impact on attention patterns. Secondly, the learned positional embeddings utilized in BERT [14] lack mathematical constraints to display periodic patterns. They are similarly added before the first model layer. Thirdly, Alibi [36] introduces a linear bias to attention scores. The linear bias is devoid of wave patterns. The remaining popular positional embeddings used in LLMs such as xPos [39] are RoPE-based variants. These variants are predominantly modified for long-context extrapolation rather than better context awareness. Therefore, they share the same shortcoming: tokens in attention trough are less focused on, thereby limiting context awareness, which is the study focus in our paper.

E Details on expert sets

Utilizing the RoPE-base searching algorithm as proposed by Chen et al. [8], Table 9 illustrates the resulting sets for different values of N.

Table 9: Searched Sets for Different N

N	**Searched Set**
3	{10,000, 18,000, 19,000}
5	{10,000, 17,500, 18,000, 19,000, 20,000}
7	{10,000, 17,500, 18,000, 19,000, 20,000, 22,500, 25,000}
9	{10,000, 13,500, 17,500, 18,000, 19,000, 20,000, 22,500, 24,000, 25,000}

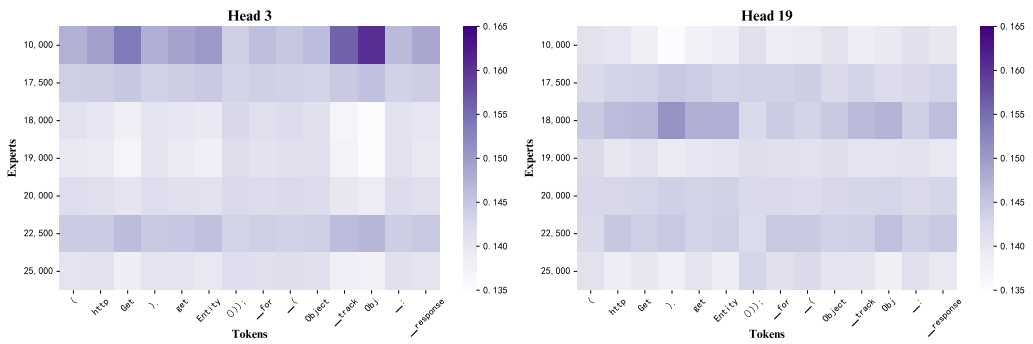

Figure 4: The routing weights across two distinct attention heads at the 27th layer in Llama-2-7B-chat. The input tokens are randomly sampled from the training data, and the attention heads under observation are also randomly selected. The horizontal axis depicts the input tokens, while the vertical axis represents experts with varying RoPE angles. Due to their distinct functions, each head dynamically chooses different experts to process individual tokens. Input text can be found in Figure 5.

F Limitations

In this paper, we introduce a plug-in module called MoICE, which is integrated into the attention heads of open-source LLMs to enhance their context awareness. One limitation is that, due to limited computational resources, we did not investigate the effectiveness of pretraining a language model with more parameters using the MoICE architecture. Furthermore, our proposed method exploits the potential for context awareness within LLMs, but it does not imbue the models with additional inherent context awareness abilities. Achieving this may necessitate more extensive data to train all model parameters.

G Broader impacts and safety issues

Our novel lightweight plug-in approach efficiently enhances the context awareness of open-source LLMs. This advancement holds great promise for enhancing the effectiveness of LLMs across diverse scenarios characterized by extensive and complex contexts, such as RAG, tool utilization, and role-playing. The safety issue of our method mainly comes from the large language models we used, as they might output toxic and biased texts, which is a common safety issue regarding LLM research.

Table 10: The mean and standard deviation of MoICE. We repeat L-eval [1] experiments 5 times with different random seeds. The randomness of MoICE results from the initialization of MoICE router when training, which causes slight differences in performance.

Method	Closed - Ended Task					Open - Ended Task		
	Coursera	QuALITY	TOEFL	SFiction	Average	wins	ties	win-rate%
Llama2-7B-Chat [42]	36.77 ± 0.00	38.12 ± 0.00	55.02 ± 0.00	60.16 ± 0.00	47.52 ± 0.00	68.00 ± 0.00	117.00 ± 0.00	34.94 ± 0.00
+ MoICE	39.65 ± 0.32	41.88 ± 0.27	56.28 ± 0.21	64.84 ± 0.00	50.66 ± 0.05	89.00 ± 1.00	117.20 ± 1.48	40.77 ± 0.20
Mistral-7B-Instruct-8k [20]	45.20 ± 0.00	44.06 ± 0.00	62.08 ± 0.00	61.72 ± 0.00	53.27 ± 0.00	71.00 ± 0.00	105.00 ± 0.00	34.11 ± 0.00
+ MoICE	48.08 ± 0.24	46.73 ± 0.27	65.35 ± 0.81	62.18 ± 1.19	55.59 ± 0.16	85.00 ± 1.10	115.20 ± 2.05	39.39 ± 0.21

… Here's how to retrieve the top tracks of an artist: ```java import org.apache.http.client.methods.HttpGet; import org.apache.http.impl.client.HttpClientBuilder; public class SpotifyAPI { private static final String ARTIST_ID = "spotify_artist_id"; public static void main(String[] args) throws Exception { String accessToken = "your_access_token"; HttpGet httpGet = new HttpGet(String.format("https://api.spotify.com/v1/artists/%s/top-tracks?country=US", ARTIST_ID)); httpGet.setHeader(HttpHeaders.AUTHORIZATION, "Bearer " + accessToken); JSONObject response = new JSONObject(EntityUtils.toString(HttpClientBuilder.create().build().execute(httpGet).getEntity())); for (Object trackObj : response…

Figure 5: The input text in Figure 4. To clearly display, we only show part of the input text, where the text with a yellow background corresponds to the decoded tokens.

```
"eb098018-bdb5": "970cbed8-3665",
"0a9d957f-2256": "be09fd63-4dfa",
"e2b49af9-d0e3": "c5ed6251-085d",
"8ece1451-05e1": "2d5932f7-acd8",
"eb2f4a8d-e0b7": "e0acbc2c-d478",
"0c8c0695-dd3c": "086d71cb-35c0",
"79a1c002-4ba6": "e69f5f62-250e",
"b0c1c9df-c13f": "3ce6b12e-6223",
"ee17cc77-6342": "41c410e1-776c",
"483f6a4d-9aa4": "3711356c-6df1",
"ee17cc77-6342": "41c
```

Figure 6: The input prompt example in Section 5.5. We use 10 key-value pairs as examples in prompt, which includes a query key. We insert the query key-value pair in different positions of examples (In the prompt example above, the query key is inserted in the 9th position). The model's task is to find the value corresponding to the query key and output it, which evaluates its ability of context awareness.

Table 11: The ablation study on the auxiliary loss of MoICE. To assess the impact of this loss term, we perform an ablation experiment on two LLMs by removing it from Eq. 10. The results show a significant drop in performance, highlighting the positive impact of the auxiliary loss.

Method	Coursera	QuALITY	TOEFL	SFiction	Average
Llama2-7B-chat [42]					
MoICE w/o aux loss	39.83	41.58	56.13	62.50	50.01
MoICE w/ aux loss	39.83	42.08	56.13	64.84	50.72
Mistral-7B-Instruct-8k [20]					
MoICE w/o aux loss	47.67	46.04	64.68	58.59	54.25
MoICE w/ aux loss	47.82	46.53	64.68	62.50	55.38

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose an effective and efficient approach for enhancing the context awareness of LLMs.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We create a separate "Limitations" section in Appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper does not introduce theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We've shared the link to the code. We promise to open source.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We promise to open code. We have posted an anonymous code link.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the p-value in the t-test in the "Results and Analysis" paragraph of Section 4.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have conformed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Section "Broader Impacts and Safeguards."

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: See Section "Broader Impacts and Safety Issues."

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have correctly cited all the data, scripts, and models we used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have a README document for our code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing experiments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing experiments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.