

# MUA: Multiple Context Units Attention For Large Language Models

Zhiyi Li<sup>1</sup>, Yaming Yang<sup>2</sup>, Kai Shuang<sup>1\*</sup>, Yiren Chen<sup>3</sup>,

<sup>1</sup>State Key Laboratory of Networking and Switching Technology,  
Beijing University of Posts and Telecommunications

<sup>2</sup>Independent Researcher

<sup>3</sup>Peking University

{iheadx,shuangk}@bupt.edu.cn, yamingyang@alumni.pku.edu.cn, yrchen92@pku.edu.cn

## Abstract

Large language models (LLMs) are increasingly required to process inputs composed of multiple context units, such as answer options, retrieved passages, or in-context demonstrations. Effective reasoning in such scenarios demands capturing both intrinsic semantics within each unit and the aggregated semantics across units. However, recent efforts typically treat multiple units as a single entangled sequence, ignoring the distinct knowledge within isolated units, which leads to suboptimal performance. To this end, we propose MUA<sup>1</sup> (Multiple Context Units Attention), a dual-view self-attention module that computes a local view for learning unit-specific semantics and a global view to enable bidirectional interaction across units. These two views share projection matrices and are fused by a lightweight input-conditioned gate, while generation proceeds with standard causal decoding. With an optimized implementation, this design achieves efficient fusion of multiple context units and superior information utilization during input encoding at marginal overhead. Experiments on diverse multiple context units reasoning benchmarks, including multiple-choice, multi-hop QA and language understanding tasks, show that MUA consistently improves performance across model scales, exhibits increased robustness to context units reordering, and establishes new state-of-the-art results on key benchmarks.

## 1 Introduction

Large language models (LLMs) achieve strong results on many NLP tasks, such as summarization and zero-shot learning (Pu et al., 2023; Chang et al., 2023; Kojima et al., 2022). However, their application in scenarios involving multiple context units such as multiple-choice question answering (QA), retrieval-augmented generation (RAG) (Lewis et al.,

2020), and in-context learning (ICL) (Brown et al., 2020; Akyürek et al., 2022) with multiple demonstrations presents distinct challenges. Existing research reveals that LLMs are notably sensitive to the ordering of these units (Chen et al., 2024; Zhao et al., 2021; Lu et al., 2022). Furthermore, the mechanisms for effectively learning the intrinsic knowledge within each unit, while simultaneously capturing the aggregated knowledge across units, remain significantly under-explored (Ding et al., 2024).

Recent work addresses these challenges through two primary lines. The first targets order sensitivity directly (Zaheer et al., 2017; Lee et al., 2018; Pezeshkpour and Hruschka, 2023). Set-based approaches, such as SetLLM (Egressy and Stühmer, 2025), enforce units permutation invariance so that model representations remain unchanged under units reordering. However, these methods involve substantial continued pre-training costs. As demonstrated in our empirical experiments, such techniques make LLMs insensitive to order permutations at the cost of performance degradation.

The second line of work relaxes the causal attention constraints within the prompt to allow bidirectional attention. This strengthens inter-unit knowledge learning and often improves average accuracy on benchmarks involving multiple context units (Clark et al., 2018; Sap et al., 2019; Zellers et al., 2019). However, these methods treat the learning of each unit from a fully global perspective, where any unit can attend freely to all others. This can weaken the learning of intrinsic knowledge within individual units, which may harm performance in settings that require sufficient single unit understanding prior to aggregation.

In this work, we focus on this second direction, aiming to enable LLMs to learn intra-unit semantics (within each unit) and inter-unit semantics (across units) simultaneously. Effective reasoning over multiple context units depends on two com-

\*Corresponding author.

<sup>1</sup><https://github.com/iheadx/MUA>

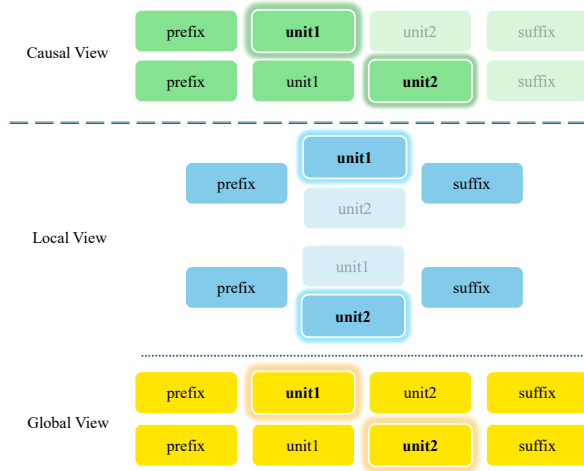


Figure 1: Attention visibility patterns for multi-context-unit prompts. Top: Causal view where visibility depends on sequence position. Bottom: Local view that isolates peer units while keeping shared Prefix/-Suffix visible, and Global view that enables prompt-bidirectional interaction across all prompt tokens.

plementary capabilities: robust cross-unit interaction for global information aggregation, and strong, self-contained intra-unit representations that preserve local semantic integrity (Beltagy et al., 2020; Zaheer et al., 2020). This necessitates attention mechanisms that balance localized unit encoding with cross-unit reasoning, rather than treating permutation invariance as the sole objective.

Learning both types of semantics within pre-trained autoregressive LLMs presents significant challenges. First, premature global interaction can dilute unit-specific semantics before they are fully formed, leading to cross-unit interference. Second, integrating complementary attention patterns within a unified decoder-only architecture requires careful design to avoid collapsing into one of two extremes: premature mixing of context units or their overly strict isolation. These inherent constraints help explain why prior work has typically addressed only one side of this trade-off.

We therefore seek a novel mechanism that supplies both unit-wise encoding and cross-unit reasoning during prompt encoding, while preserving standard causal decoding for generation.

In this work, we propose **MUA** (Multiple Context Units Attention), a **Dual-View Self-Attention** framework tailored to prompts with multiple context units (Figure 1). Standard causal self-attention imposes an inherent asymmetry: tokens in later units can attend to earlier ones, but not vice versa. This constraint limits effective information aggrega-

tion during prompt encoding (Zhao et al., 2021; Liu et al., 2024). Furthermore, processing all tokens in a single causal pass obscures the boundaries and internal structure of distinct units. Consequently, the learning of unit-specific semantics and cross-unit reasoning becomes entangled, introducing noise and suboptimal representations (Min et al., 2022).

MUA addresses these challenges by computing two complementary attention views over prompt tokens within each Transformer layer: a Local View that enforces strict unit boundaries to strengthen intra-unit encoding, and a Global View that applies bidirectional attention across all units to facilitate cross-unit comparison and aggregation. Two views share projection matrices and are dynamically fused via a lightweight, input-conditioned gate. Autoregressive generation remains unchanged and uses standard causal masking.

However, standard FlashAttention (Dao et al., 2022) do not natively support the heterogeneous masking patterns required by MUA, making a naive implementation inefficient and memory-intensive. To enable efficient decoding, we therefore reformulate MUA’s attention computation through a combination of structured mask decomposition and Fused RoPE (Su et al., 2021), which exactly preserves the intended behavior while remaining compatible with FlashAttention. This reformulation involves substantial engineering effort, yet yields near-vanilla inference efficiency and greatly reduced memory footprint.

This paper makes three contributions:

1. We unify multiple-choice QA, multi-hop reasoning, and retrieval-augmented generation under a multiple context units framework. To address this, we propose MUA, a novel dual-view self-attention module that integrates unit-wise isolation and bidirectional interaction within each Transformer layer through a lightweight, gated fusion mechanism.
2. Extensive experiments across multiple benchmarks and model scales demonstrate that MUA consistently improves both average accuracy and robustness to context-unit reordering. Notably, the gains become more pronounced as context length and the number of context units increase.
3. We develop a FlashAttention-compatible implementation of MUA through attention decomposition and shared RoPE computation,

substantially improving inference efficiency while preserving exact behavior.

## 2 Related Works

Encoder-only architectures such as BERT and its variants model structured, multi-segment inputs with fully bidirectional self-attention (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2019). In parallel, set representation learning studies inputs as unordered collections, where permutation invariance is a central requirement. Deep Sets (Zaheer et al., 2017) and Set Transformer (Lee et al., 2018) aggregate element-wise representations with symmetric operators or attention-based pooling, combining strong intra-element modeling with global aggregation. These architectures rely on bidirectional attention over the full input and therefore do not directly apply to autoregressive generation under causal constraints.

Most LLMs are decoder-only Transformers trained with causal self-attention (Vaswani et al., 2017), including GPT-style models (OpenAI et al., 2023; Yang et al., 2025). Causal masking is required for autoregressive generation (Ding et al., 2024; Yin et al., 2024), but it can be limiting for prompt encoding when the prompt contains multiple context units that benefit from bidirectional modeling. Several works modify attention masks or positional encodings within decoder-only models. Prefix language models relax causal masking inside a designated prefix region, enabling bidirectional attention during prompt encoding while preserving causal decoding (Diao et al., 2022).

Related ideas appear in instruction tuning and retrieval-augmented generation, where prompt tokens and generated tokens are treated differently (Katz et al., 2024). This separation can improve prompt representations without changing the decoding objective.

For multiple context units prompts, Segment-Based Attention Masking (MAS) relaxes the prompt mask to allow bidirectional visibility, strengthening cross-unit interaction (Katz et al., 2024). A different direction treats the units as a set and enforces permutation invariance to remove order effects (Zhao et al., 2021; Eggeessy and Stühmer, 2025). SetLLM follows this route and typically relies on extra pre-training, which increases training cost (Eggeessy and Stühmer, 2025).

Prior work points to a trade-off in decoder-only models with multiple context units. Global interac-

tion helps comparison and aggregation, but early cross-unit mixing can degrade unit representations. This paper explores an attention design that coordinates complementary local and global views within a unified decoder-only layer.

## 3 Method

This section details the Multiple Context Units Attention (MUA) mechanism. Motivated by the need to balance unit-specific encoding with cross-unit reasoning, MUA comprises three key elements: (1) a Local View for intra-unit attention, (2) a Global View for inter-unit attention, and (3) a lightweight adaptive gate that dynamically fuses the two views.

### 3.1 Overview

We consider prompts consisting of a shared prefix, multiple context units, and a shared suffix, as illustrated in Figure 1. Modern LLMs, built on decoder-only Transformers, treat such prompts as a single unidirectional sequence and encode them using causal self-attention. However, this single causal pass often leads to premature cross-unit mixing, where representations from different units blend before unit-level semantics are fully formed. This can impair both intra-unit interpretation and inter-unit aggregation.

To address this, we propose to decouple the learning of unit-wise representations and cross-unit semantics through two distinct attention mechanisms, whose outputs are then adaptively fused. Furthermore, we employ different positional encoding strategies for each attention type. For learning self-contained unit representations, we enforce strict isolation by resetting positional IDs at unit boundaries (the Reset strategy). For cross-unit communication, we allow bidirectional attention between units while maintaining the original sequential order of positions (the Sequential or Seq strategy).

Based on this design, MUA computes a Local view and a Global view in parallel over prompt tokens and fuses them with an input-conditioned gate. We integrate MUA by substituting the standard self-attention mechanism in each Transformer layer with this dual-view module, while keeping all other layer components unchanged. The three core components of MUA are detailed below.

### 3.2 Preliminaries

Let the layer input be  $H \in \mathbb{R}^{T \times d}$ . We denote  $\text{Pos}(\cdot; p)$  as the operation of applying position ids  $p$

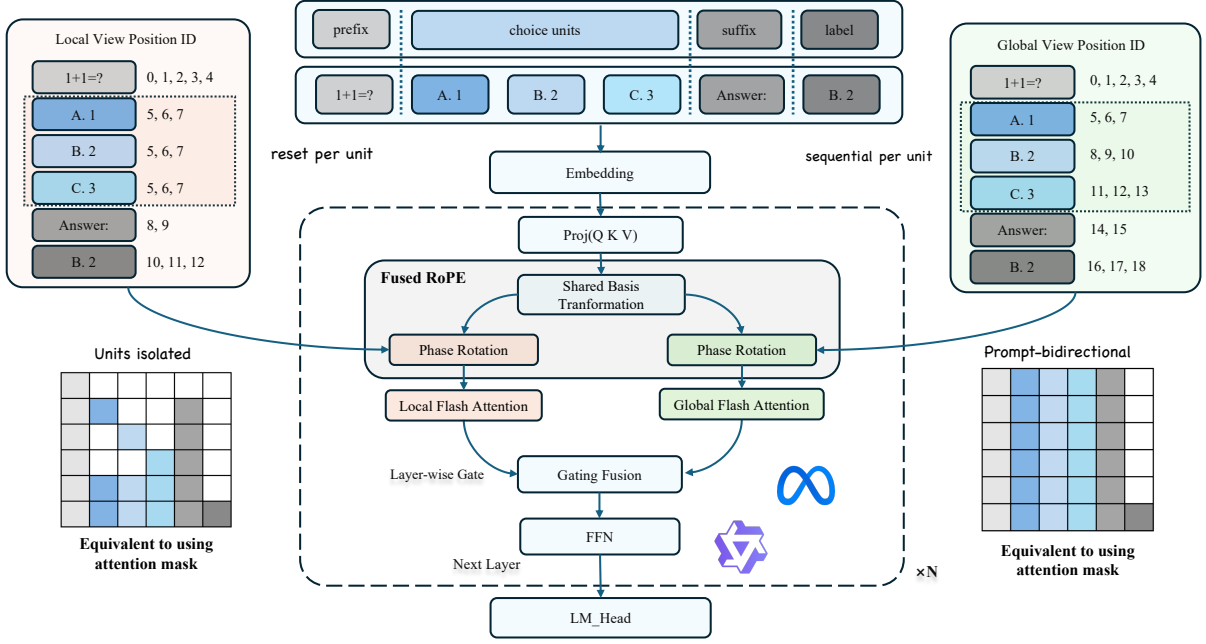


Figure 2: Conceptual overview of the Dual-View Self-Attention architecture. The input sequence is processed through two complementary paths: a **Local View** that enforces isolation between context units to extract independent intra-unit semantics, and a **Global View** that enables full interaction for comparative inter-unit reasoning. These two information streams are dynamically fused via an adaptive gating mechanism.

to a projected vector (e.g., as in RoPE). The generic multi-head attention with an explicit attention mask  $M$  and position ids  $p$  is expressed as:

$$Q = \text{Pos}(HW_Q; p), \quad (1)$$

$$K = \text{Pos}(HW_K; p), \quad (2)$$

$$\text{Attn}(H; M, p) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_h}} + M\right)(HW_V), \quad (3)$$

$$O = \text{Attn}(H; M, p)W_O. \quad (4)$$

As shown in Figure 2, MUA computes two complementary attention outputs over the prompt tokens: a Local View for intra-unit encoding and a Global View for inter-unit interaction. While the two views share the same projection matrices ( $W_Q, W_K, W_V, W_O$ ), they employ distinct attention masks and positional encoding strategies.

### 3.3 Local View: Intra-Unit Attention

The Local View aims to strengthen unit-wise encoding by isolating context units. We compute the output  $O_{\text{local}}$  using a block-structured mask  $M_{\text{local}}$  and reset position ids  $p_{\text{local}}$ :

$$O_{\text{local}} = \text{Attn}(H; M_{\text{local}}, p_{\text{local}})W_O \quad (5)$$

In  $M_{\text{local}}$ , tokens attend bidirectionally within the same context unit and to the shared prefix/suffix, while attention between different units is blocked. Concurrently, we employ position ids  $p_{\text{local}}$ , which are reset for each unit. This combination forces the model to focus on intra-unit structure and local semantics without being distracted by the global absolute positions of other units.

### 3.4 Global View: Inter-Unit Attention

Complementing the local view, the Global View facilitates aggregated reasoning across units—a necessity for tasks such as summarizing evidence from multiple paragraphs or selecting relevant examples. We compute the global output  $O_{\text{global}}$  by allowing unrestricted interaction across the prompt:

$$O_{\text{global}} = \text{Attn}(H; M_{\text{global}}, p_{\text{global}})W_O \quad (6)$$

Here,  $M_{\text{global}}$  applies a fully bidirectional mask over the prompt region, enabling tokens from all context units (as well as the prefix and suffix) to attend to one another. We utilize position ids  $p_{\text{global}}$ , which are sequential across the entire input, to capture relative structural relationships and distance between units. Note that for generated tokens, autoregressive masking is preserved in both views to

ensure compatibility with standard decoder-only language modeling.

### 3.5 Adaptive Gated Fusion

The Local and Global views provide complementary signals: the former captures unit-wise semantics, while the latter facilitates cross-unit aggregation. We integrate these views using a lightweight gate conditioned on the pre-attention hidden state. Formally, for layer  $\ell$  and token  $t$ , the fused output is computed as:

$$g_t^{(\ell)} = \sigma\left(W_{\text{gate}}^{(\ell)} h_t^{(\ell)} + b_{\text{gate}}^{(\ell)}\right), \quad (7)$$

$$o_t^{(\ell)} = g_t^{(\ell)} \odot o_{t,\text{local}}^{(\ell)} + \left(1 - g_t^{(\ell)}\right) \odot o_{t,\text{global}}^{(\ell)} \quad (8)$$

where  $\sigma$  is the sigmoid function and  $\odot$  denotes element-wise multiplication. This corresponds to implementing  $W_{\text{gate}}^{(\ell)}$  as a single linear layer applied to the attention input.

We adopt this input-conditioned gating mechanism for its stability and efficiency. By avoiding complex attention-style interactions (e.g., head-wise reweighting), the gate dynamically modulates the mixture of Local and Global information based on token content without complicating optimization. Note that this dual-view mechanism is applied exclusively to the input context; during token generation, the model reverts to standard causal masking.

### 3.6 Efficient FlashAttention-Compatible Implementation

Naively implementing MUA incurs substantial overhead due to heterogeneous masking patterns and distinct positional encodings in the Local and Global views, preventing direct use of optimized attention kernels. To improve practicality, we develop an exact FlashAttention-compatible reformulation of MUA.

First, we avoid redundant rotary computations between the two views. Instead of independently applying RoPE (Su et al., 2021) twice, we first compute a shared basis transformation of the projected queries and keys, and then apply lightweight view-specific phase rotations corresponding to the Local and Global position ids. This reformulation preserves exact behavior while reducing repeated tensor reads and element-wise operations.

Second, we reformulate the Local View attention to eliminate dense  $O(S^2)$  masks. We observe

that only tokens inside context units require non-standard visibility, whereas prefix and suffix regions naturally follow standard causal attention. Based on this property, Local View computation is decomposed into: (1) a standard causal FlashAttention pass over the full sequence, and (2) lightweight per-unit corrections computed only on valid sub-contexts (prefix, current unit, and suffix) using bidirectional FlashAttention. The resulting unit-wise outputs overwrite the corresponding context-unit regions, yielding behavior mathematically equivalent to the original structured masking without explicit mask materialization. The Global View directly applies bidirectional FlashAttention over prompt tokens.

Finally, to minimize overhead when many context units are present, we batch per-unit computations using variable-length FlashAttention kernels. Overall, our implementation preserves the exact behavior of MUA while substantially improving efficiency. On Llama3.2-1B with input length 4096, our optimized implementation reduces prefill latency by 81%, decoding latency by 10%, and peak VRAM usage by 61%.

## 4 Experiments

### 4.1 Experimental Setup

**Models.** We evaluate our method across a range of model sizes: LLAMA-3.2-1B/3B, LLAMA-3.1-8B, and QWEN-3-14B. All models are initialized from their publicly released instruction-tuned checkpoints. Our implementation modifies only the attention mechanism, replacing standard self-attention with MUA, while leaving all other components, including layer normalization, feed-forward networks, and residual connections unchanged.

**Tasks.** We evaluate on three categories of tasks that inherently involve reasoning over multiple context units: multiple-choice QA (ARC-Challenge) (Clark et al., 2018), multi-hop QA (2WikiMultihopQA, HotpotQA) (Ho et al., 2020; Yang et al., 2018), which we use as a substitute for ICL and RAG tasks, providing the full context rather than retrieved snippets or few-shot examples, and broad language understanding (MMLU) (Hendrycks et al., 2020).

**Compared Methods.** We compare the following model variants, which share identical backbone parameters and differ only in their attention computation: Base(The standard fine-tuned base-

line using causal self-attention.), Bitune(Kopiczko et al., 2024), SetLLM(Egressy and Stühmer, 2025), MAS(Katz et al., 2024) and MUA.

## 4.2 Evaluation Metric

To eliminate the randomness incurred by different units order, we randomly sample  $K$  different orderings of the context units for each example during evaluation. Here we use  $K = 5$  unless otherwise specified. To further assess the robustness of different methods to context unit reordering, we also compute strictly correctness across all  $K$  samples (5-Adv), specifically, the prediction should be all correct for the sampled  $K$  orderings, otherwise, the prediction is incorrect, we name this metric 5-Adv.

## 4.3 Main Results

As shown in Table 1, MUA consistently outperforms both the standard fine-tuned baseline (Base) and the strong segment-based attention masking baseline (MAS) across all evaluated models and tasks. Notably, it achieves maximum improvements of 10.37% on the MMLU language understanding benchmark, 4.3% on the multi-hop QA dataset (HotpotQA), and 11.95% on the ARC-Challenge multiple-choice task. These results demonstrate the effectiveness of MUA in enhancing reasoning across diverse multi-context unit scenarios.

## 4.4 Ablation Studies

We conduct ablation studies using the 1B and 3B LLaMA-3.2 models to examine the contribution of each core component in MUA. Specifically, we evaluate the following model variants:

**MUA:** Our complete model with both the Local View and Global View, fused via an adaptive gate.

**w/o G-View:** Ablates the Global View, retaining only the Local View for strict intra-unit encoding.

**w/o L-View:** using only the Global View.

**w/o Gate:** Replaces the adaptive gate with a simple summation of the two attention outputs.

The performance of the fine-tuned baseline (Base) is included for comparison.

Table 2 presents the results of our ablation study.

**View Ablations.** We first isolate the contributions of the two views. Retaining only the Local view (w/o G-View) yields strong robustness (high 5-Adv and Cons.), as unit-wise isolation minimizes interference caused by permutation. However, this

comes at the cost of accuracy, as cross-unit reasoning is restricted. Conversely, retaining only the Global view (w/o L-View) improves accuracy over the baseline by enabling direct cross-unit interaction, yet its robustness underperform the full model. This indicates that global mixing alone fails to preserve stable unit semantics. The full MUA model effectively combines both strengths: the Local view anchors robust unit representations, while the Global view facilitates the interaction necessary for high accuracy.

**Gating Ablation.** The "MUA w/o Gate" variant replaces the input-conditioned fusion with a simple summation of the Local and Global attention outputs. The resulting performance drop demonstrates that a static combination of signals is insufficient. Instead, the model benefits significantly from the gate’s ability to dynamically modulate the balance between Local and Global information based on token content and layer depth.

## 5 Analysis

### 5.1 Robustness under Context Reordering

We evaluate robustness to context unit reordering by sampling five random permutations per example. As shown in Table 3, MUA consistently improves not only average accuracy (Acc) but also the robustness-oriented metrics (5-Adv and Cons.) relative to both the base model and MAS. This trend persists across diverse model scales and backbones, suggesting that the performance gains are agnostic to model size. Notably, MUA outperforms MAS, which relies on fully bidirectional global mixing across all reported settings. This indicates that MUA produces significantly more stable predictions under context reordering. These results validate our core design hypothesis: the Local view mitigates permutation-induced interference by preserving unit-wise semantics, while the Global view maintains the cross-unit interactions essential for high accuracy.

### 5.2 Attention Sensitivity to Context Unit Permutations

To investigate the stability of different attention mechanisms, we analyze how attention distributions shift under context unit reordering. Specifically, we examine whether the model consistently attends to semantically relevant units regardless of their absolute positions.

| Models                              | HotpotQA     | MMLU         | 2WikiMultihopQA | ARC-e        | ARC-c        | AVG          |
|-------------------------------------|--------------|--------------|-----------------|--------------|--------------|--------------|
| <i>Llama3.2 1B</i>                  | 33.83        | 48.16        | 64.63           | 67.00        | 49.30        | 52.58        |
| <i>Llama3.2 1B<sub>SetLLM</sub></i> | 34.85        | 49.15        | 76.17           | 79.55        | 53.61        | 58.67        |
| <i>Llama3.2 1B<sub>Bitune</sub></i> | 31.43        | 45.87        | 73.59           | 74.79        | 52.65        | 55.67        |
| <i>Llama3.2 1B<sub>MAS</sub></i>    | 35.00        | 47.06        | 86.50           | 72.70        | 55.50        | 60.25        |
| <i>Llama3.2 1B<sub>MUA</sub></i>    | <b>36.53</b> | <b>50.22</b> | <b>87.82</b>    | <b>81.36</b> | <b>61.26</b> | <b>63.44</b> |
| <i>Llama3.2 3B</i>                  | 38.20        | 54.54        | 86.69           | 83.00        | 70.60        | 66.61        |
| <i>Llama3.2 3B<sub>SetLLM</sub></i> | 38.04        | 58.64        | 86.86           | 88.55        | 74.16        | 69.25        |
| <i>Llama3.2 3B<sub>Bitune</sub></i> | 37.22        | 59.36        | 85.27           | 88.68        | 75.94        | 69.29        |
| <i>Llama3.2 3B<sub>MAS</sub></i>    | 39.07        | 61.05        | 89.85           | 89.40        | 75.10        | 70.89        |
| <i>Llama3.2 3B<sub>MUA</sub></i>    | <b>40.49</b> | <b>61.70</b> | <b>91.11</b>    | <b>91.20</b> | <b>78.16</b> | <b>72.53</b> |
| <i>Llama3.1 8B</i>                  | 37.02        | 56.99        | 88.33           | 88.80        | 78.00        | 69.83        |
| <i>Llama3.1 8B<sub>SetLLM</sub></i> | 39.45        | 63.83        | 89.67           | 91.12        | 84.13        | 73.01        |
| <i>Llama3.1 8B<sub>MAS</sub></i>    | 40.27        | 66.38        | 90.52           | <b>92.20</b> | 81.70        | 74.21        |
| <i>Llama3.1 8B<sub>MUA</sub></i>    | <b>41.32</b> | <b>67.36</b> | <b>91.40</b>    | 90.95        | <b>84.98</b> | <b>75.20</b> |
| <i>Qwen3 14B</i>                    | 40.57        | 77.66        | 90.86           | <b>97.64</b> | 93.09        | 79.96        |
| <i>Qwen3 14B<sub>MAS</sub></i>      | 40.49        | 77.34        | 90.60           | 97.43        | 92.58        | 79.69        |
| <i>Qwen3 14B<sub>MUA</sub></i>      | <b>41.42</b> | <b>78.29</b> | <b>91.75</b>    | 97.60        | <b>93.86</b> | <b>80.58</b> |

Table 1: The Acc under different models and methods

| Model               | HotpotQA     |              |              | MMLU         |              |              |
|---------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                     | Acc          | 5-Adv.       | Cons.        | Acc          | 5-Adv.       | Cons.        |
| <b>LLAMA-3.2-1B</b> |              |              |              |              |              |              |
| Base                | 33.82        | 26.97        | 51.25        | <u>48.16</u> | 26.97        | 51.44        |
| MUA                 | <b>36.53</b> | <b>33.34</b> | <u>71.92</u> | <b>50.22</b> | 38.95        | 66.50        |
| w/o G-View          | 31.98        | <u>31.37</u> | <b>95.00</b> | 45.90        | <b>45.26</b> | <b>97.61</b> |
| w/o L-View          | 35.00        | 28.67        | 55.76        | 47.06        | 28.68        | <u>66.50</u> |
| w/o Gate            | <u>35.12</u> | 31.29        | 68.59        | 45.57        | 33.75        | 62.91        |
| <b>LLAMA-3.2-3B</b> |              |              |              |              |              |              |
| Base                | 38.20        | 31.88        | 58.88        | 54.54        | 41.92        | 64.48        |
| MUA                 | <b>40.49</b> | <b>37.41</b> | <u>75.02</u> | <b>61.70</b> | 52.24        | 73.10        |
| w/o G-View          | 37.32        | <u>36.89</u> | <b>96.19</b> | 58.68        | <b>58.20</b> | <b>98.46</b> |
| w/o L-View          | <u>39.07</u> | 33.73        | 39.27        | 61.05        | 46.25        | 62.75        |
| w/o Gate            | 38.56        | 35.54        | 72.32        | 57.97        | 47.71        | 72.50        |

Table 2: Ablation results on HotpotQA and MMLU for attention-view variants.

We extract attention maps from the final Transformer layer, averaging scores across all heads. For each query token in the suffix and the generated label tokens, we aggregate the attention weights assigned to tokens within each context unit. These unit-level scores are then normalized to sum to one for visualization.

Figure 3 presents several representative examples. Ideally, the model should attend strongly to the units containing the evidence for the correct answer. In the original order, the baseline attends to the relevant unit and predicts correctly. However, after shuffling, the baseline’s attention drifts to an irrelevant unit, leading to an incorrect prediction.

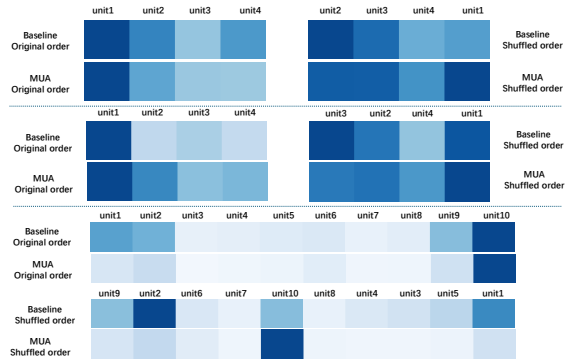


Figure 3: Attention maps under two orders on MMLU and hotpotQA. The baseline flips from correct (original) to wrong (shuffled), while MUA stays correct.

In contrast, MUA preserves a stable attention distribution focused on the relevant unit across both permutations, resulting in robust and correct predictions. We attribute this stability to the Local view, which anchors unit-level representations and mitigates interference from positional changes.

### 5.3 Performance across Context Lengths

We further investigate the scalability of our method by analyzing performance trends with respect to input length and the number of context units. We analyze the impact of input complexity using the Full-Wiki (context length) and Distractor (unit count) sub-tasks of HotpotQA with LLAMA-3.2-3B.

Figure 4 demonstrates that MUA consistently

| Models                           | HotpotQA     |              |              | MMLU         |              |              | 2WikiMultihopQA |              |              |
|----------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------------|--------------|--------------|
|                                  | Acc          | 5-Adv.       | Cons.        | Acc          | 5-Adv.       | Cons.        | Acc             | 5-Adv.       | Cons.        |
| <i>Llama3.2 1B</i>               | 33.83        | 26.97        | 51.25        | 48.16        | 26.97        | 51.44        | 64.63           | 55.66        | 78.25        |
| <i>Llama3.2 1B<sub>MAS</sub></i> | 35.00        | 28.67        | 55.76        | 47.06        | 28.68        | 44.66        | 86.5            | 79.86        | 87.23        |
| <i>Llama3.2 1B<sub>MUA</sub></i> | <b>36.53</b> | <b>33.34</b> | <b>71.92</b> | <b>50.22</b> | <b>38.95</b> | <b>66.50</b> | <b>87.82</b>    | <b>85.43</b> | <b>95.09</b> |
| <i>Llama3.2 3B</i>               | 38.20        | 31.88        | 58.88        | 54.54        | 41.92        | 64.48        | 86.69           | 82.15        | 90.59        |
| <i>Llama3.2 3B<sub>MAS</sub></i> | 39.07        | 33.73        | 62.89        | 61.05        | 46.25        | 62.75        | 89.85           | 86.73        | 94.03        |
| <i>Llama3.2 3B<sub>MUA</sub></i> | <b>40.49</b> | <b>37.41</b> | <b>75.02</b> | <b>61.70</b> | <b>52.24</b> | <b>71.30</b> | <b>91.11</b>    | <b>90.20</b> | <b>97.85</b> |
| <i>Llama3.1 8B</i>               | 37.02        | 31.49        | 59.15        | 56.99        | 43.45        | 64.51        | 88.33           | 84.66        | 92.37        |
| <i>Llama3.1 8B<sub>MAS</sub></i> | 40.27        | 35.75        | 66.28        | 66.38        | 53.87        | 69.32        | 90.52           | 87.91        | 94.80        |
| <i>Llama3.1 8B<sub>MUA</sub></i> | <b>41.32</b> | <b>38.61</b> | <b>78.14</b> | <b>67.36</b> | <b>59.16</b> | <b>78.60</b> | <b>91.40</b>    | <b>90.60</b> | <b>98.20</b> |
| <i>Qwen3 14B</i>                 | 40.57        | 33.68        | 56.34        | 77.66        | 68.20        | 78.42        | 90.86           | 88.31        | 94.42        |
| <i>Qwen3 14B<sub>MAS</sub></i>   | 40.49        | 34.57        | 60.43        | 77.34        | 68.28        | 78.85        | 90.60           | 87.13        | 92.99        |
| <i>Qwen3 14B<sub>MUA</sub></i>   | <b>41.42</b> | <b>38.00</b> | <b>72.25</b> | <b>78.29</b> | <b>71.75</b> | <b>84.30</b> | <b>91.75</b>    | <b>90.61</b> | <b>97.38</b> |

Table 3: Robustness under context-unit reordering. Acc is the average accuracy over five random permutations for each example; 5-Adv (Adversarial Order) is the fraction of examples answered correctly under all five permutations; Cons. is the fraction with identical predictions across the five permutations (higher is better).

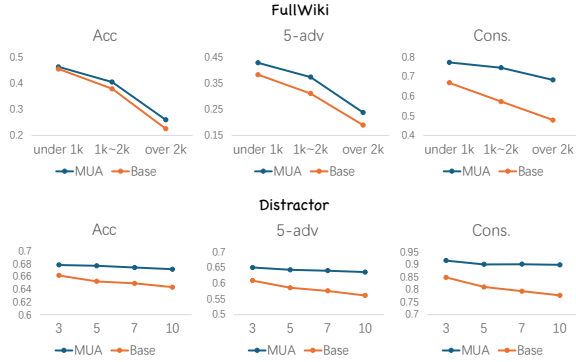


Figure 4: Performance on HotpotQA under different input settings. **Top: FullWiki** results across context-length buckets. **Bottom: Distractor** results across different numbers of answer options. We report Acc, 5-Adv., and Cons., comparing the baseline and MUA.

outperforms the baseline across all context length intervals on FullWiki and varying numbers of candidate options on Distractor. These improvements are evident in both accuracy and robustness metrics (5-Adv. and Cons.). Notably, the performance gap between MUA and the baseline widens as context length and option count increase. This trend highlights MUA’s effectiveness in handling complex reasoning tasks that demand the integration of information across multiple text segments.

#### 5.4 Efficiency Analysis

Table 4 compares training FLOPs between base model and MUA. Across model sizes, MUA adds modest compute overhead; the accuracy gains do not require a large increase in training cost.

In Table 5, results shows that after our inference optimization, the inference efficiency of MUA has already approached that of standard LLMs with

| Model | Base                  | MUA                   |
|-------|-----------------------|-----------------------|
| 1B    | $2.95 \times 10^{18}$ | $3.15 \times 10^{18}$ |
| 3B    | $6.19 \times 10^{18}$ | $6.77 \times 10^{18}$ |
| 8B    | $1.64 \times 10^{19}$ | $1.77 \times 10^{19}$ |

Table 4: Training compute cost (FLOPs) on HotpotQA for models from 1B to 8B.

| Len.  | Bitune | SetLLM | MUA   |
|-------|--------|--------|-------|
| 1024  | 23.49  | 20.02  | 21.34 |
| 2048  | 23.58  | 19.61  | 21.19 |
| 4096  | 23.30  | 19.20  | 21.37 |
| 8192  | 23.53  | 21.09  | 21.33 |
| 16384 | 23.33  | OOM    | 21.42 |

Table 5: The decode latency(ms/token) analysis on Llama3.2-1B. The Nvidia A40 GPU used during testing has 48 GB of VRAM, but the SetLLM runs out of memory when the input text length reaches 16384.

FlashAttention, and at the same time, the growth of GPU memory usage has also slowed down significantly.

## 6 Conclusion

We introduced MUA, a dual-view mechanism for decoder-only LLMs that balances intra-unit semantic formation and inter-unit aggregation. Modeling isolated Local and bidirectional Global views, MUA improves reasoning performance while maintaining standard autoregressive decoding.

Experiments demonstrate that MUA improves performance across reasoning tasks and model scales, while exhibiting robustness to context-unit ordering and outperforming masking-based and permutation-invariant baselines. Furthermore, we show that MUA admits an efficient FlashAttention-

compatible reformulation, enabling practical long-context inference through structured attention decomposition without dense attention masks.

## Limitations

Our approach relies on the explicit partitioning of the input into distinct context units and shared prompt components. In scenarios where unit boundaries are ambiguous or task-specific formatting varies significantly, constructing the Local mask and unit-local position ids may require heuristic preprocessing. Furthermore, MUA is designed to balance local and global interactions to improve robustness, rather than enforcing strict permutation invariance; consequently, it does not theoretically guarantee identical outputs under all reorderings.

## Acknowledgments

This work was supported by National Key R&D Program of China(Grant No. 2024YFF0618700)

## References

- Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. [What learning algorithm is in-context learning? Investigations with linear models](#). *arXiv e-prints*, arXiv:2211.15661.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The Long-Document Transformer](#). *arXiv e-prints*, arXiv:2004.05150.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language Models are Few-Shot Learners](#). *arXiv e-prints*, arXiv:2005.14165.
- Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2023. [BoookScore: A systematic exploration of book-length summarization in the era of LLMs](#). *arXiv e-prints*, arXiv:2310.00785.
- Xinyun Chen, Ryan A. Chi, Xuezhi Wang, and Denny Zhou. 2024. [Premise Order Matters in Reasoning with Large Language Models](#). *arXiv e-prints*, arXiv:2402.08939.
- 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 e-prints*, arXiv:1803.05457.
- Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. [FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness](#). *arXiv e-prints*, arXiv:2205.14135.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Shizhe Diao, Wangchunshu Zhou, Xinsong Zhang, and Jiawei Wang. 2022. [Write and Paint: Generative Vision-Language Models are Unified Modal Learners](#). *arXiv e-prints*, arXiv:2206.07699.
- Nan Ding, Tomer Levinboim, Jialin Wu, Sebastian Goodman, and Radu Soricut. 2024. [Causallm is not optimal for in-context learning](#). In *International Conference on Learning Representations*, volume 2024, pages 12574–12585.
- Beni Egressy and Jan Stühmer. 2025. [Set-LLM: A Permutation-Invariant LLM](#). *arXiv e-prints*, arXiv:2505.15433.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. [Measuring Massive Multitask Language Understanding](#). *arXiv e-prints*, arXiv:2009.03300.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Shahar Katz, Liran Ringel, Yaniv Romano, and Lior Wolf. 2024. [Segment-Based Attention Masking for GPTs](#). *arXiv e-prints*, arXiv:2412.18487.
- Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.
- Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. 2024. [Bitune: Leveraging Bidirectional Attention to Improve Decoder-Only LLMs](#). *arXiv e-prints*, arXiv:2405.14862.
- Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiosek, Seungjin Choi, and Yee Whye Teh. 2018. [Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks](#). *arXiv e-prints*, arXiv:1810.00825.

- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. **BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension**. *arXiv e-prints*, arXiv:1910.13461.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks**. *arXiv e-prints*, arXiv:2005.11401.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. **Lost in the middle: How language models use long contexts**. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. *arXiv e-prints*, arXiv:1907.11692.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. **Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. **Rethinking the role of demonstrations: What makes in-context learning work?** In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 181 others. 2023. **GPT-4 Technical Report**. *arXiv e-prints*, arXiv:2303.08774.
- Pouya Pezeshkpour and Estevam Hruschka. 2023. **Large Language Models Sensitivity to The Order of Options in Multiple-Choice Questions**. *arXiv e-prints*, arXiv:2308.11483.
- Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. **Summarization is (Almost) Dead**. *arXiv e-prints*, arXiv:2309.09558.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. **Social iqa: Commonsense reasoning about social interactions**. In *EMNLP*.
- Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. **RoFormer: Enhanced Transformer with Rotary Position Embedding**. *arXiv e-prints*, arXiv:2104.09864.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. **Attention Is All You Need**. *arXiv e-prints*, arXiv:1706.03762.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. **Qwen3 Technical Report**. *arXiv e-prints*, arXiv:2505.09388.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. **HotpotQA: A dataset for diverse, explainable multi-hop question answering**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Qingyu Yin, Xuzheng He, Xiang Zhuang, Yu Zhao, Jianhua Yao, Xiaoyu Shen, and Qiang Zhang. 2024. **StableMask: Refining Causal Masking in Decoder-only Transformer**. *arXiv e-prints*, arXiv:2402.04779.
- Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. **Big Bird: Transformers for Longer Sequences**. *arXiv e-prints*, arXiv:2007.14062.
- Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. 2017. **Deep Sets**. *arXiv e-prints*, arXiv:1703.06114.
- 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, Florence, Italy. Association for Computational Linguistics.
- Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. **Calibrate Before Use: Improving Few-Shot Performance of Language Models**. *arXiv e-prints*, arXiv:2102.09690.

## A Hyperparameters of Training

We use a unified set of hyperparameters for supervised fine-tuning (SFT) across all models (Base, MAS and MUA variants from 1B to 8B). We use AdamW with a learning rate of  $2 \times 10^{-5}$ , batch size 64, and train for 3 epochs with weight decay 0.01 and warmup ratio 0.03. Unless otherwise specified,

| Model | Length | Units | Base (ms) | MUA (ms) |
|-------|--------|-------|-----------|----------|
| 1B    | 1024   | 4     | 35.70     | 54.62    |
| 1B    | 1024   | 10    | 35.18     | 63.10    |
| 1B    | 2048   | 4     | 67.73     | 100.64   |
| 1B    | 2048   | 10    | 68.59     | 107.86   |
| 1B    | 4096   | 4     | 137.45    | 205.97   |
| 1B    | 4096   | 10    | 137.69    | 213.67   |
| 3B    | 1024   | 4     | 90.94     | 135.79   |
| 3B    | 1024   | 10    | 91.42     | 148.94   |
| 3B    | 2048   | 4     | 177.36    | 252.48   |
| 3B    | 2048   | 10    | 178.06    | 267.00   |
| 3B    | 4096   | 4     | 348.84    | 523.39   |
| 3B    | 4096   | 10    | 350.12    | 545.95   |

Table 6: Prefill latency of the standard attention baseline and MUA under different sequence lengths and numbers of context units.

| $N$  | Prefix= $N$ , Suffix=256 | Prefix=256, Suffix= $N$ |
|------|--------------------------|-------------------------|
| 512  | 153.73                   | 155.24                  |
| 1024 | 206.71                   | 208.74                  |
| 2048 | 339.57                   | 344.85                  |
| 4096 | 605.78                   | 615.61                  |

Table 7: Prefill latency (ms) after exchanging the prefix and suffix lengths. Prefix and suffix placement has little effect on latency.

we use a fixed random seed 42 and run training on NVIDIA H20 GPUs. For inference, all models use greedy decoding. For the Qwen3-14B model, we use LoRA adapters during SFT, with rank 32, alpha 64 and a learning rate of  $1 \times 10^{-4}$ . The gate parameters use learning rate of  $2 \times 10^{-5}$ . Other hyperparameters remain the same as above.

## B Additional Efficiency Analysis

### B.1 Prefill Latency

Table 6 compare the standard SDPA baseline and the fused MUA implementation on Llama-3.2-1B and Llama-3.2-3B. We vary the sequence length and the number of context units.

### B.2 Effect of Long Shared Prefixes and Suffixes

The Local View exposes the shared prefix and suffix to every unit. We therefore measure whether the placement of a long shared region introduces asymmetric overhead. We use Llama-3.2-3B with four units of 100 tokens each. As shown in Table 7, for a fixed value of  $N$ , exchanging a long prefix and a long suffix produces nearly identical prefill latency.

Table 8 shows that when the prefix and suffix grow together, latency increases with total

| Prefix | Suffix | Total length | Prefill (ms) |
|--------|--------|--------------|--------------|
| 512    | 512    | 1424         | 184.57       |
| 1024   | 1024   | 2448         | 303.81       |
| 2048   | 2048   | 4496         | 589.65       |
| 4096   | 4096   | 8592         | 1274.00      |

Table 8: Prefill latency as the shared prefix and suffix grow together.

| Layer | Dim. std. | Specialized | Local | Global | Balanced |
|-------|-----------|-------------|-------|--------|----------|
| 0     | 0.015     | 0.0%        | 0.0%  | 0.0%   | 100.0%   |
| 1     | 0.036     | 0.2%        | 0.2%  | 0.0%   | 99.8%    |
| 4     | 0.041     | 1.4%        | 0.8%  | 0.6%   | 98.6%    |
| 8     | 0.049     | 4.3%        | 1.9%  | 2.4%   | 95.7%    |
| 10    | 0.056     | 7.4%        | 3.1%  | 4.3%   | 92.6%    |
| 13    | 0.064     | 11.8%       | 6.2%  | 5.6%   | 88.2%    |
| 15    | 0.053     | 6.1%        | 3.4%  | 2.7%   | 93.9%    |

Table 9: Dimension-level gate statistics. “Specialized” is the fraction of dimensions with  $g > 0.6$  or  $g < 0.4$ .

input length, as expected for dense attention. The FlashAttention implementation avoids explicit dense-mask materialization but does not change the theoretical complexity of dense attention.

## C Gate Behavior Analysis

We analyze the learned gate of Llama-3.2-1B MUA using 100 ARC-Challenge test examples. In Eq. (8), larger gate values assign more weight to the Local View, while smaller values assign more weight to the Global View. Because the gate is dimension-wise, Table 9 reports the standard deviation across hidden dimensions and the proportion of dimensions that are Local-dominant ( $g > 0.6$ ), Global-dominant ( $g < 0.4$ ), or balanced ( $0.4 \leq g \leq 0.6$ ).

As shown in Table 9, most dimensions in shallow layers remain balanced. The specialized-dimension ratio grows in deeper layers and peaks at Layer 13, with similar proportions favoring the Local and Global Views. These observations suggest that deeper layers learn a more differentiated mixture of intra-unit and cross-unit information. They should not be interpreted as assigning a fixed semantic role to every hidden dimension.

## D Dataset Details

### D.1 MMLU (Massive Multitask Language Understanding)

#### Data Statistics:

- Training set: 99,842 samples

- Validation set: 1,531 samples
- Test set: Contains multiple subjects, totaling approximately 14,000+ samples

#### **Prompt Format Example:**

Prefix: The cyclic subgroup of  $Z_{24}$  generated by 18 has order  
Options: ["4", "8", "12", "6"]  
Suffix: The correct answer is:  
Label: " 4"

## **D.2 HotPotQA**

### **Data Statistics(fullwiki):**

- Training set: 90,447 samples
- Validation set: 7,405 samples

### **Data Split(distractor):**

- Validation set (3 options): 7,405 samples
- Validation set (5 options): 7,405 samples (same questions, different number of options)
- Validation set (7 options): 7,405 samples (same questions, different number of options)

#### **Prompt Format Example:**

Prefix: Please answer questions based on the following text.  
Options: ["Woodson, Arkansas (CDP description...)", "Ed Wood (biography...)", "...", "Scott Derrickson (biography...)"]  
Suffix: Were Scott Derrickson and Ed Wood of the same nationality?  
Label: "No"

## **D.3 ARC-C**

### **Data Statistics:**

- Training set: 1,119 samples
- Test set: 1,172 samples

#### **Prompt Format Example:**

Prefix: Juan and LaKeisha roll a few objects down a ramp. They want to see which object rolls the farthest. What should they do so they can repeat their investigation?  
Options: ["Put objects in groups.", "Change the height of ramp.", "Choose different objects to roll.", "Record the details of the investigation."]   
Suffix: The correct answer is:  
Label: " Record the details of the investigation ."

## **D.4 2WikiMultiHopQA**

### **Data Statistics:**

- Training set: 167,000+ samples
- Validation/Test sets: 12,600+ samples

#### **Prompt Format Example:**

Prefix: Please answer questions based on the following text.  
Options: ["Franco Rossi (director)...", "Jeannette: The Childhood of Joan of...", "Jean Pasquerel (Jean Pasquerel( c. 1400) was ...", ...]  
Suffix: Which film has the director who died earlier, Il Seduttore or The Trial Of Joan Of Arc?  
Label: "The Trial Of Joan Of Arc"

## **E Unified Input Sequence Structure**

In the dual attention model, the input sequence is constructed as:

[BOS] + prefix  
+ option<sub>1</sub> + option<sub>2</sub> + ... + option<sub>N</sub>  
+ suffix