

Article

An Empirical Study on Enhancing Large Language Models for Long-Term Conversations in Korean

Hongjin Kim ^{1,2} , Jeonghyun Kang ¹, Yeajin Jang ^{1,3}, Yujin Sim ^{1,4} and Harksoo Kim ^{1,*} 

¹ Department of Artificial Intelligence, Konkuk University, 120, Neungdong-ro, Gwangjin-gu, Seoul 05029, Republic of Korea; jin3430@konkuk.ac.kr (H.K.); dpwls258@konkuk.ac.kr (Y.J.); simuzin@konkuk.ac.kr (Y.S.)

² Electronics and Telecommunications Research Institute, 218, Gajeong-ro, Yuseong-gu, Daejeon 34129, Republic of Korea

³ Korea Electronics Technology Institute, 22, Daewangpangyo-ro 712beon-gil, Bundang-gu, Seongnam-si 13488, Gyeonggi-do, Republic of Korea

⁴ SweetK, 401, Simin-daero, Dongan-gu, Anyang-si 14057, Gyeonggi-do, Republic of Korea

* Correspondence: nlprkim@konkuk.ac.kr

Abstract

Large language models (LLMs) have shown strong performance in open-domain dialogue, yet they continue to struggle with long-term multi-session conversations (MSC), particularly in non-English languages such as Korean. In this work, we present a comprehensive empirical study on enhancing Korean MSC capabilities of LLMs through dataset construction, memory modeling, and parameter-efficient fine-tuning. We introduce an extended Korean MSC dataset that explicitly distinguishes between persona memory (long-term user attributes) and episode memory (short-term, event-driven information), enabling more effective memory management across sessions. Using this dataset, we evaluate LLM performance on three core MSC tasks: session summarization, memory update, and response generation. Our experiments reveal that Korean MSC is intrinsically more challenging than English MSC and that memory update and response generation require substantial reasoning ability. To address these challenges, we compare LoRA, DPO, MoE, CPT, Layer Tuning, and neuron-level tuning methods. Results consistently show that neuron tuning, guided by a novel language-specific neuron identification method based on activation scores and entropy, achieves superior performance and robustness, particularly in continual learning settings. Overall, our findings highlight neuron-level adaptation as an effective and interpretable approach for improving long-term conversational ability in low-resource languages.

Keywords: multi-session conversation; long-term conversation; memory update; large language models; language-specific neurons



Academic Editors: Minoru Sasaki
and Francisco García-Sánchez

Received: 23 February 2026

Revised: 14 March 2026

Accepted: 16 March 2026

Published: 25 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Open-domain conversation systems aim to generate appropriate responses to user input. The advent of pre-trained language models (PLMs) such as BART [1], combined with the availability of large-scale, high-quality dialogue datasets, has led to significant advancements in the response generation capabilities of open-domain chatbots. However, these systems continue to face challenges in multi-session (i.e., long-term) conversations, where retaining and utilizing information from previous sessions remains difficult. In particular, generating engaging responses that effectively integrate past and current context

remains an open problem. This difficulty arises from the need for several key capabilities in multi-session chat (MSC) settings. In this study, we define three essential abilities for MSC:

- **Session Summarization:** To generate appropriate responses, models must be able to recall information from previous sessions or user-provided inputs. While storing entire conversations is possible, summarization has been shown to be a more effective and efficient approach [2].
- **Memory Update:** Models must be capable of refining and updating user-related information, such as emotional state, location, or recent events, which can dynamically change over time in MSC scenarios [3].
- **Response Generation:** When responding to user utterances, models should reason about the relevance of stored memory in relation to the current input and determine whether past information should be utilized for response generation [4].

As outlined above, the integration of these complex abilities presents a significant challenge for generating coherent and engaging responses in multi-session conversations. Although large language models (LLMs) have demonstrated promising capabilities in MSC, these advancements have primarily been observed in resource-rich languages such as English and Chinese. In contrast, a noticeable performance gap persists between high-resource and low-resource languages [5–7]. Furthermore, a recent study [8] revealed that even top open and closed LLMs show lower performance in multi-turn conversations than in single-turn. In this study, we first show that various LLMs exhibit degraded long-term conversational capabilities, particularly in Korean, compared to English, consistent with findings from prior work on other low-resource languages [9]. Despite the significant capability gap between English and Korean, efforts to address this disparity in long-term conversations remain largely unexplored. To investigate whether recent fine-tuning approaches can enhance Korean long-term conversation capabilities, we conduct a comprehensive set of experiments. Specifically, we apply low-rank adaptation (LoRA) [10], direct preference optimization (DPO) [11], mixture-of-experts (MoE) [12], continual pre-training (CPT) [13], and Layer Tuning [14] techniques for instruction-tuning on Korean datasets for session summarization, memory update, and MSC. We evaluate the effectiveness of each method and analyze their impact on Korean long-term conversational performance. Our results reveal that while these fine-tuning approaches yield improvements on individual tasks, they suffer from catastrophic forgetting in continual learning settings where all three tasks are learned sequentially. To address this limitation, we draw inspiration from recent research on language-specific neurons [9,15] and identify Korean-specific neurons in various LLMs. We demonstrate that selectively tuning these neurons not only enhances the models' long-term conversational capabilities in Korean [16] but also exhibits robust performance in continual learning settings.

To fine-tune models for MSC in Korean, appropriate datasets are essential. While ref. [2] introduced a multi-session chat dataset for English—including session-level summaries to support dialogue summarization—there remains a significant scarcity of long-term open-domain conversational datasets for Korean. Moreover, collecting MSC datasets is resource-intensive, requiring substantial time and financial investment. To promote research on long-term open-domain conversations in Korean, we construct new datasets for session summarization and memory update tailored to Korean MSC. For the session summarization and memory update dataset, we build upon the dataset introduced in our prior work [3], augmenting it with additional annotations to better support memory update modeling in MSC systems. Unlike previous studies [2,3,17], we explicitly distinguish between persona memory and episode memory during annotation. This separation is intended to support more efficient memory management as session histories grow. Given that memory capacity is limited, we argue—drawing inspiration from how humans manage

memory—that long-term conversational systems should differentiate between information that must persist over time (persona memory) and information that can be updated or discarded (episode memory). Defining and annotating these two memory types is a key distinction of our approach compared to existing MSC datasets. In summary, our main contributions are as follows:

1. To address the capability gap between English and Korean, we conduct comprehensive experiments to enhance the long-term conversational abilities of LLMs in Korean.
2. We construct Korean session summarization and memory update datasets. Unlike existing MSC datasets, our dataset differentiates between persona and episode memory to reflect long-term versus short-term information.
3. Experimental results demonstrate the effectiveness of our methods in long-term conversation settings. Additionally, our MSC dataset—annotated with distinct memory types—supports the generation of more engaging and contextually appropriate responses.
4. We demonstrate that selectively tuning Korean-specific neurons outperforms existing fine-tuning approaches and exhibits robust performance in continual learning settings where other methods suffer from catastrophic forgetting.

2. Related Work

2.1. MSC Dataset

To address long-term open-domain conversations, ref. [2] constructed and released an MSC dataset and demonstrated the effectiveness of both retrieval-augmented generative (RAG) models, which search for and retrieve relevant utterances from previous sessions, and read–write memory-based models, which summarize and store previous session conversations. Ref. [18] introduced the LOCCO (Long-term Chronological Conversations) dataset, a benchmark designed to assess how LLMs retain and utilize information over extended temporal spans. Their analysis reveals that LLM memory representations exhibit temporal decay and category-specific retention patterns. However, this dataset only covers English, which is a resource-rich language. Ref. [17] build a Korean MSC dataset, which is named CareCall-Memory (<https://github.com/naver-ai/carecall-memory> (accessed on 1 January 2024)) based on the CareCall dataset (<https://github.com/naver-ai/carecall-corpus> (accessed on 1 January 2024)). Unfortunately, the CareCall-memory dataset may not be suitable for chat data since it was generated under the premise of a nurse calling senior citizens to provide care (i.e., its fundamental source is from speech). As a result, the CareCall-memory dataset includes utterances that are not appropriate for chat data, such as “I’m calling from the administrative welfare center” and “Can you hear me well?”. Another Korean MSC dataset, known as KMSC, was constructed by AI Hub (<https://www.aihub.or.kr/> (accessed on 1 January 2024)). However, as noted in our previous work [3], the KMSC dataset suffers from low-quality session summaries and does not consider the memory update process. To address these limitations, we introduced KEEM (Keep Emotion and Essential Memory) [3], a generation-based memory update dataset for long-term conversation that enables models to update user emotion and episodic information. In this study, we further annotate the KEEM dataset to explicitly distinguish between **persona memory** and **episode memory**. We demonstrate that this distinction facilitates more effective memory updating by LLMs and, in turn, leads to improved response generation. We provide further analysis and empirical results in Section 5.

2.2. Memory-Augmented Dialogue Systems

Recent work on memory-augmented dialogue systems has explored external memory modules to enable long-term personalization and conversational continuity. While such approaches extend LLMs beyond their fixed context windows, many rely on rigid memory

granularity (e.g., turn-level or session-level storage) and static retrieval strategies. To address these limitations, Reflective Memory Management (RMM) [19] introduces a dual-reflection framework that integrates hierarchical memory summarization with adaptive retrieval refinement through reinforcement learning. In a complementary direction, the SHARE dataset [20] explores relational shared memories between interlocutors, constructed from movie scripts capturing both explicit persona information and implicit shared experiences. The accompanying EPISODE framework demonstrates that modeling shared memories can improve dialogue engagement and sustainability. While these approaches advance memory architectures for English dialogue systems, they do not address the unique challenges of non-English languages or the explicit separation of long-term and short-term memory. Our work differs by focusing on Korean MSC and introducing a distinct persona-episode memory framework that supports dynamic memory updates across sessions.

2.3. Language-Specific Neuron

Ref. [15] investigated the existence of language-specific neurons in LLMs and proposed a method called *language activation probability entropy* (LAPE) to identify them. Their findings suggest that the ability of an LLM to process a specific language can be influenced by a relatively small subset of neurons, thereby supporting the existence of language-specific neurons. Additionally, they found that these neurons are predominantly located in the bottom and top layers of LLMs. According to their analysis, the lower layers primarily convert inputs from various languages into a shared semantic space—typically aligned with a high-resource language such as English—while the upper layers map this semantic content (processed through the middle layers) into language-specific output tokens. In a similar vein, ref. [9] examined the multilingual processing pipeline of LLMs. Their study confirmed that LLMs initially convert multilingual inputs into English, perform reasoning in English, and integrate multilingual knowledge using self-attention and feed-forward components, respectively. Finally, the models generate responses in the original input language. To further investigate this mechanism, ref. [9] proposed *parallel language-specific neuron detection* (PLND), a method for identifying the importance of neurons in both attention and feed-forward components without requiring labeled data or parameter modifications. Ref. [13] aimed to mitigate the significant performance gap in mathematical reasoning between English and Korean by identifying Korean-specific neurons. Their findings indicate that neurons in the shallow layers, which are responsible for internal translation from Korean inputs to English representations, are particularly critical for effective fine-tuning. Inspired by these model interpretability-based studies on language-specific neurons, we identify Korean-specific neurons in various LLMs and selectively fine-tune them to enhance the MSC capabilities of LLMs in Korean.

3. KEEM Dataset

Data Construction

As we describe in Sections 1 and 2, there remains a scarcity of MSC datasets in Korean. Therefore, we collected the KEEM dataset utilizing the KMSC dataset [3]. To ensure topic diversity in dialogues, we choose the KMSC dataset from the AI Hub. The KMSC dataset covers 13 diverse topics, including individuals and relationships, education, climate, and beauty and health, among others (see Table 1 for more details). It consists of four-session dialogues, each accompanied by a summary. However, the dataset was not constructed with memory update mechanisms in mind; instead, summaries from previous sessions are simply concatenated and used as memory for the following session. Additionally, like many existing datasets, the KMSC summaries frequently omit information related to the user's emotional state or its cause. For example, in the utterance “I work as a designer.

Recently, I sometimes feel embarrassed because of past work mistakes,” the corresponding summary—“I’m a designer and I sometimes feel embarrassed these days”—includes the emotion but omits the underlying cause. To build the KEEM dataset—which incorporates both emotional states and their causal context into generation-based memory updates—we created new summaries using dialogues from the KMSC dataset, with the assistance of GPT-4.0 [21] (While GPT-4.0 was used for dataset construction, it was the state-of-the-art model at the time. More importantly, the quality of the KEEM dataset is independent of the generation model, as it has been rigorously validated through comprehensive human evaluation in our previous work [3]). GPT has shown strong capabilities across various NLP tasks, including automatic dataset construction [22], and its efficiency and reliability have been validated by multiple studies [23]. We used GPT to enrich session summaries by embedding the user’s emotional expressions and the reasons behind them. Then, it was further prompted to generate an updated summary that combines information from both the current and previous sessions. Figure 1 provides an overview of the KEEM dataset construction process. Table 2 reports statistics for the KEEM datasets. In this table, “Session 1–*n*” indicates data containing *n* sequential sessions. For instance, “Session 1–4” includes the first four sessions, rather than referring to only the fourth. However, unlike previous works such as [2,17], we further instruct GPT 5.0 to separate persona and episode summaries after the KEEM dataset construction (Full instruction is provided in Appendix A). This is done to distinguish between long-term and relatively short-term information. We define the persona memory as long-term memory, which encompasses user self-introductions, personal details, preferences, possessions, family or friends, pets, and aspirations. Conversely, we define episode memory as short-term memory, which includes episodes that users recount from their daily lives. By differentiating between long-term and short-term information in this manner, the management of memories can be more effectively achieved when conversations continue. Table 3 shows an example of persona and episode memory.

Table 1. Topic statistics of KMSC dataset.

Topic	Count	Ratio
Individuals & Relationships	12,783	15.98%
Entertainment	11,063	13.83%
Beauty & Health	9038	11.30%
Society	7328	9.17%
Work & Job	6997	8.75%
Arts & Culture	6508	8.13%
Education	6288	7.86%
Food	4130	5.15%
Climate	3975	4.97%
Traffic	3795	4.75%
House	3489	4.37%
Fashion	235	0.29%

Table 2. Statistics of the original KMSC dataset and our KEEM dataset. Avg. denotes the average.

Attribute	KMSC			KEEM		
	Session 1–2	Session 1–3	Session 1–4	Session 1–2	Session 1–3	Session 1–4
Total Episodes	40,000	20,000	20,000	2006	1560	1005
Total Utterances	980,919	731,705	953,405	61,354	70,974	59,847
Total Memory sentences	567,176	395,808	498,497	33,972	31,846	23,148
Avg. Length of Utterances	16.98	17.13	17.22	17.02	17.26	17.32
Avg. Number of Utterances	30.65	45.73	59.59	30.58	45.49	59.54
Avg. Number of Memory Sentences	17.72	24.73	31.15	16.93	20.41	23.03

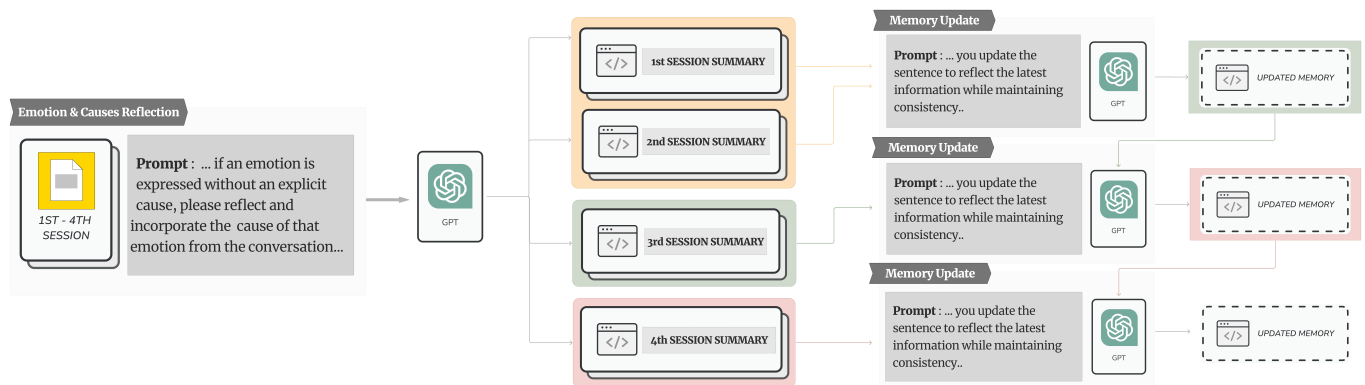


Figure 1. Process of KEEM dataset creation. We reused a figure from our previous publication [3] (CC BY 4.0 license), with full attribution provided in the figure caption.

Table 3. Examples of persona memory and episode memory.

Persona Memory	Episode Memory
My name is Andrew	I'm traveling in Europe
I'm a university student	I've been to Porto and Lisbon
I have a sister	I'm currently in Barcelona
I'm a dog person	I feel calm and peaceful these days and enjoy my vacation
...	...

4. Method

As mentioned in Section 1, we adopt various fine-tuning methods to explore ways to enhance the long-term conversational capabilities of LLMs in Korean. In this section, we describe recent fine-tuning approaches, including LoRA, DPO, MoE, and Neuron Tuning. We conduct comprehensive empirical experiments using these techniques across three tasks: session summarization, memory update, and response generation. Let $P = \{p_0, p_1, \dots, p_n\}$ denote the instruction tokens for a given task, where n is the number of tokens in the instruction. Let $X = \{x_0, x_1, \dots, x_m\}$ and $Y = \{y_0, y_1, \dots, y_l\}$ represent the input and output tokens for the task, where m and l denote the number of tokens in the input and output, respectively. We define the input as the concatenation of the instruction and the task-specific input, denoted as $I = [P; X]$. The objective is to generate the output Y (e.g., a session summary, an updated memory, or a response) conditioned on the instruction and input:

$$Y = \prod_{i=1}^{|Y|} P_W(y_i | y_{<i}, [P; X]) \quad (1)$$

Here, $P_W(\cdot | \cdot)$ denotes the probability of generating the next token given the previous tokens and the input I , and W represents the parameters of the model. The notation $y_{<i} = \{y_1, y_2, \dots, y_{i-1}\}$ refers to the prefix of the output sequence up to position $i - 1$.

4.1. LoRA

Continual Pre-Training (CPT) of LLMs is both time- and resource-intensive. As a result, many researchers have adopted LoRA as an effective and efficient alternative for fine-tuning LLMs. Furthermore, CPT poses the risk of degrading a model's parametric knowledge due to updates to the original pre-trained weights. In contrast, LoRA preserves the original weights of the LLM by introducing additional trainable parameters, thereby mitigating the risk of catastrophic forgetting during task-specific fine-tuning. The loss function minimized during LoRA fine-tuning is defined as

$$L = \frac{1}{|D|} \sum_{d \in D} L(d, \Phi_0, \Delta\Phi) \quad (2)$$

where D is the dataset, d is a single data instance, L is the cross-entropy loss function, Φ_0 represents the frozen pre-trained weights of the LLM, and $\Delta\Phi$ denotes the trainable parameters introduced by LoRA. LoRA implements low-rank matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, where $r \ll \min(d, k)$. The trainable projection is defined as

$$h = W_0x + BAx = (W_0 + BA)x \quad (3)$$

Here, W_0 is the original frozen weight matrix, and x is the input vector. We apply LoRA with two training strategies: (1) Task-independent, where a separate LoRA is trained for each individual task, and (2) Continual learning, where a single LoRA is continually trained across all three tasks. The results and analysis of these strategies are presented in a later section.

4.2. DPO

Reinforcement Learning (RL) is a promising approach for enhancing the capabilities of LLMs. In particular, Reinforcement Learning with Human Feedback (RLHF) has proven effective in aligning LLMs with human preferences and values [11]. Applying RLHF typically requires reward annotations for different outputs given the same input. These annotations are used to train a reward model, which in turn guides the LLM to generate outputs that are more aligned with human preferences by assigning scores to candidate responses. DPO was recently proposed as an efficient alternative that replicates the benefits of RLHF without requiring a separate reward model. To adopt DPO in our experiments, we only require a preference dataset indicating which output is preferred under a given input. As described in Section 3, we constructed the KEEM dataset using the KMSC dataset as a foundation. In this process, we utilized dialogues from KMSC and generated new session summaries and memory updates. Based on this, we automatically construct a preference dataset by pairing the original KMSC summaries with the improved KEEM summaries under the same input dialogues. The resulting dataset allows us to train models to prefer the KEEM-generated summaries over those from KMSC. We note that for the memory update and response generation tasks, preference datasets cannot be constructed, as the session summaries and updated memories differ between the KEEM and KMSC datasets. Therefore, DPO is applied exclusively to fine-tune the session summarization task. We apply DPO to fine-tune LLMs using preference pairs (x, y^+, y^-) . The objective is to encourage the model to assign a higher likelihood to the preferred session summary y^+ over the less preferred session summary y^- , relative to a reference model. The loss function is defined as

$$\mathcal{L}_{DPO}(\theta) = -\log \sigma \left(\beta \cdot \left(\log \frac{P_\theta(y^+|x)}{P_{ref}(y^+|x)} - \log \frac{P_\theta(y^-|x)}{P_{ref}(y^-|x)} \right) \right) \quad (4)$$

Here, $P_\theta(y|x)$ denotes the conditional probability of response y given input x , computed by the trainable policy model with parameters θ . $P_{ref}(y|x)$ represents the probability estimated by the reference model, typically a frozen snapshot of the initial model before DPO training. $\beta > 0$ is a temperature scaling factor that controls the sharpness of preference enforcement, and $\sigma(\cdot)$ denotes the sigmoid function.

4.3. MoE

MoE has been adopted to fine-tune neural networks for multi-task learning [12,24], and more recently, it has been applied to the top layers of LLMs to fine-tune them on target tasks. In this study, we also apply MoE to the top layer of LLMs and train the expert modules on three tasks: session summarization, memory update, and response generation, while keeping the original LLM weights frozen. Previous studies have explored the effectiveness of maintaining a shared expert that remains active across all training instances, demonstrating its benefits [25,26]. Motivated by these findings, we incorporate a shared expert in our MoE setup to investigate whether it contributes to performance improvements across the three tasks. This approach is based on the intuition that these tasks—being subtasks of MSC—may benefit from learning shared underlying capabilities. Given the hidden representation $h \in \mathbb{R}^d$, the MoE output is computed as

$$\text{MoE}(h) = \sum_{i \in \mathcal{T}_k(h)} w_i \cdot \text{Expert}_i(h), \quad w_i = \frac{\exp(g_i)}{\sum_{j \in \mathcal{T}_k(h)} \exp(g_j)} \quad (5)$$

where

- $\mathcal{T}_k(h)$ denotes the set of top- k expert indices selected by the gating network based on the input h ,
- $\text{Expert}_i(h)$ is the output of the i -th expert network,
- w_i is the normalized gating weight for expert i ,
- g_i is the gating score for expert i , computed by the gating network.

4.4. Neuron Tuning

Before tuning specific neurons, it is necessary to identify and select which neurons should be tuned. Ref. [15] proposed LAPE (Language Activation Probability Entropy) to identify language-specific neurons in LLMs. Specifically, they input texts in various languages into the LLM and compute the activation probability for the j -th neuron in the i -th layer when processing texts in a specific language k :

$$p_{i,j}^k = \mathbb{E} (\mathbb{I}(\text{activation_function}(\tilde{h}^i W_1^i) > 0) \mid \text{language } k) \quad (6)$$

Here, $\mathbb{I}$ denotes the indicator function. In [15], they estimate the activation probability as the likelihood that the activation of a neuron exceeds zero. Subsequently, they obtain the distribution $p_{i,j} = (p_{i,j}^1, \dots, p_{i,j}^k, \dots, p_{i,j}^l)$ which consists of the activation probabilities of the (i, j) -th neuron across l different languages. Finally, LAPE is computed as the entropy of this distribution, quantifying the degree of language specificity for each neuron:

$$\text{LAPE}_{i,j} = - \sum_{k=1}^l \hat{p}_{i,j}^k \log(\hat{p}_{i,j}^k) \quad (7)$$

where $\hat{p}_{i,j}^k$ denotes the normalized probability over languages (We apply L1 normalization). A low LAPE score indicates that the neuron is language-specific, as it tends to be active for only one or a few languages while remaining inactive for others. However, LAPE [15] exhibits certain limitations in cases with low entropy values. Specifically, LAPE may yield a small value when a neuron is highly likely to be activated in language k but consistently unlikely in other languages. Unfortunately, it can also produce a similarly small value when the neuron is only weakly activated in language k while remaining inactive in others. In the latter case, the neurons identified by LAPE may have limited influence on language k . To mitigate this limitation, we modify and extend the LAPE method to more accurately identify language-specific neurons. Another limitation of LAPE arises during

the conversion of activation scores into probabilities. In this step, activation scores greater than zero are mapped to one, and all others to zero, which fails to capture the magnitude of activation. Consequently, the resulting probabilities do not reflect the true influence of each neuron. To address this issue, we propose a new approach that identifies neurons based on activation scores and entropy, rather than activation probabilities.

Activation Scores of Attention Module

For the input h_{i-1} at the i -th layer and the m -th head, the query, key, and value vectors in the attention module are defined as follows:

$$\begin{aligned} q^{(i,m)} &= W^{Q,(i,m)} h_{i-1}, \\ k^{(i,m)} &= W^{K,(i,m)} h_{i-1}, \\ v^{(i,m)} &= W^{V,(i,m)} h_{i-1} \end{aligned} \quad (8)$$

Note that modern transformer architectures (e.g., Llama, Qwen, Gemma) apply no activation function after these projections. The activation score of the j -th neuron for an input c is defined as the absolute value of the projection output:

$$\begin{aligned} a_j^{(i,m,Q)}(c) &= \left| \left(W^{Q,(i,m)} h_{i-1}(c) \right)_j \right|, \\ a_j^{(i,m,K)}(c) &= \left| \left(W^{K,(i,m)} h_{i-1}(c) \right)_j \right|, \\ a_j^{(i,m,V)}(c) &= \left| \left(W^{V,(i,m)} h_{i-1}(c) \right)_j \right| \end{aligned} \quad (9)$$

where $|\cdot|$ denotes the absolute value, capturing the magnitude of neuron activation regardless of sign.

Activation Scores of the FFN

Modern LLMs such as Llama, Qwen, and Gemma employ the SwiGLU activation function in their feed-forward networks. The FFN is formulated as

$$FFN(h) = W^{down} \cdot (\sigma(W^{gate} h) \odot W^{up} h) \quad (10)$$

where σ denotes the SiLU (Swish) activation function and $\odot$ represents element-wise multiplication. We measure the activation score at the intermediate representation after the gated activation (i.e., the “post” activation in TransformerLens terminology):

$$\tilde{h}_i = \sigma(W^{gate} h_{i-1}) \odot W^{up} h_{i-1} \quad (11)$$

The activation score of the j -th neuron in the FFN for an input c is then

$$a_j^{(i,mlp)}(c) = \left| (\tilde{h}_i(c))_j \right| \quad (12)$$

For an input corpus C_k in language k , the mean activation score is computed as

$$\text{ActScore}_j^{(i,M)}(k) = \mathbb{E}_{c \sim C_k} [a_j^{(i,M)}(c)], \quad M \in \{Q, K, V, \text{mlp}\} \quad (13)$$

The entropy of each neuron's activation distribution across languages is defined as

$$H_j^{(i,M)} = - \sum_{k=1}^L \tilde{a}_j^{(i,M)}(k) \log \tilde{a}_j^{(i,M)}(k),$$

$$\tilde{a}_j^{(i,M)}(k) = \frac{\text{ActScore}_j^{(i,M)}(k)}{\sum_{k'} \text{ActScore}_j^{(i,M)}(k')} \quad (14)$$

Identification of Language-Specific Neurons

Finally, a neuron is identified as language-specific for language k if it satisfies both of the following conditions among all neurons across layers: (i) its activation score for language k falls within the top 1%, and (ii) its entropy lies within the bottom 25%. Formally,

$$N_{\text{lang}} = \left\{ j \in \mathcal{N} \mid \text{ActScore}_j^{(M)}(k) \in \text{Top-1\%}, \right. \\ \left. H_j^{(M)} \in \text{Bottom-25\%} \right\} \quad (15)$$

Unlike Equations (13) and (14), which are computed per layer, Equation (15) is applied across all neurons in all layers to detect the top 1% neurons. Consequently, without layer-wise normalization of the activation scores $\text{ActScore}_j^{(i,M)}(k)$, neurons may become concentrated in layers with higher overall activation magnitudes [27]. In this study, we compare and analyze the results obtained with and without normalization to examine whether neurons are evenly distributed across layers or concentrated in high-influence layers.

In this study, since we focus on Korean, we select only those neurons that exhibit high activation scores $\text{ActScore}_j^{(i,M)}(\text{KOR})$ and low entropy $H_j^{(i,M)}$ values for fine-tuning the LLMs. We further fine-tune the identified Korean-specific neurons using the following training objective:

$$\mathcal{L} = - \sum_{t=1}^T \log p(x_t \mid x_{<t}; \theta_{\text{frozen}} + \Delta_S) \quad (16)$$

where S denotes the set of Korean-specific neurons.

4.5. Continual Pre-Training

We perform continual pre-training on LLMs using the KEEM dataset to enhance Korean MSC capabilities and to elicit reasoning abilities in Korean, following the approach of [13].

4.6. Specific Layer Tuning

Inspired by knowledge editing techniques [14], we identify and fine-tune the layer containing the highest concentration of Korean-specific neurons to improve Korean MSC capabilities.

4.7. Task

4.7.1. Session Summarization

Given a query q (i.e., instructions) and a session dialogue $D = \{u_1, u_2, u_3, \dots, u_n\}$ consisting of n utterances, our goal is to generate a session summary (i.e., memory) $M = [S^p, S^e]$. Here, S^p denotes the **persona memory** and S^e denotes the **episode memory**. Both components are generated jointly in this process.

4.7.2. Memory Update

Formally, given a previous session's memory M_{prev} and a current session's memory M_{cur} , we train LLMs to generate an updated memory M^U . The objective is to retain and revise important user information or events across sessions. For example, suppose the user stated "I'm travelling in Europe" in a previous session (e.g., a few months ago) and later says "I'm back! I'm in Korea now" in the current session. In this case, we expect the updated memory to contain "I have been to Europe", thereby preserving relevant historical information while incorporating the new context. This approach addresses limitations of existing memory update methods, which typically remove conflicting information from the previous memory when inconsistencies arise between M_{prev} and M_{cur} [17]. In the same example, the conventional method would delete "I'm travelling in Europe" and retain only "I'm back! I'm in Korea now", causing the model to lose an important event from the user's history.

4.7.3. Response Generation

Finally, we train LLMs to generate coherent and engaging responses in the MSC environment, conditioned on the updated memory M^U and the current session dialogue D_{cur} . Through this process, the LLM learns when and what information in M^U to attend to, and how to generate an appropriate response to the user's most recent utterance.

It is important to note that we train these tasks using *LoRA*, *DPO*, *MoE*, *CPT*, *Layer Tuning*, and *Neuron tuning*, respectively. We then evaluate which method is most effective for enhancing Korean MSC capabilities.

5. Experiments

As mentioned in Section 1, we first demonstrate that LLMs encounter greater difficulty in processing Korean MSC tasks than English ones. We then train the models using various methods and evaluate their effectiveness.

5.1. Experimental Setup

We used EXAONE 4 [28], Gemma 3 [29], Llama 3.1 [30], Qwen 2.5 [31], and Qwen 3 [32] models to fine-tune for our experiments. For fine-tuning LLMs, we used our KEEM dataset.

5.2. Implementation Details

All experiments were conducted on eight NVIDIA A100 GPUs (80 GB each). For EXAONE 4 and Qwen 3, which support reasoning mode, we follow the optimal configuration: temperature = 0.6, top- p = 0.95, and top- k = 20. For other models that do not provide a separate reasoning mode, we set the temperature to 0.0, and both top- p and top- k to 1. For LoRA, we set the LoRA α to 64 and the dropout rate to 0.1. The rank parameter r (i.e., the LoRA dimension size) was set to 128. We used a batch size of 32, with 4 gradient accumulation steps, and a learning rate of 2×10^{-4} for all the tuning methods. For continual pre-training, we conduct training on 8 A100 GPUs over 3 epochs, using a context length of 4096 tokens and a warm-up ratio of 0.01. The optimizer applies a weight decay of 0.01, and the peak learning rate is set to 2×10^{-5} , following an inverse square root decay schedule. Training is performed in FP16 precision using DeepSpeed and FlashAttention. For the MoE method, we add MoE layers on top of the frozen LLM and train only the expert modules. We conduct a hyperparameter search over the number of experts $\in \{3, 4, 6, 8, 12\}$ and top- k routing $\in \{1, 2, 3, 4\}$. The best performance is achieved with 8 experts and top-2 routing, which we adopt for all reported results. We employ a shared expert architecture: one expert is designated as a shared expert that remains always activated, while the top-2 experts are dynamically selected from the remaining 7 routed experts based on softmax

gating scores. Each expert is implemented as a two-layer feed-forward network with a hidden dimension of the backbone model. The gating network computes a softmax distribution over experts, and the top-2 experts with the highest gating scores are selected for each input token. For neuron activation measurement, we utilized the TransformerLens library for models it supports, and PyTorch (torch==2.7.1) forward hooks for other models. Specifically: Attention Q/K/V: We capture the output of linear projection layers ('q_proj', 'k_proj', 'v_proj'), which do not include activation functions; FFN (MLP): We capture the intermediate activation after the gated activation function, corresponding to the "post" hook point in TransformerLens. This represents $\sigma(W^{gate} \cdot h) \odot W^{up} \cdot h$ in SwiGLU-based architectures. The activation score is computed as the absolute value of these outputs, averaged across all tokens in the input sequence. (Our code and detailed instructions will be made available on GitHub.)

5.3. Human Evaluator Details

For manual evaluation, we recruit five native Korean speakers with familiarity in chatbot systems as human evaluators. To ensure unbiased assessment, evaluators are not informed of the dataset construction procedures or the specific methodologies used in our approach and the baselines.

5.4. Evaluation of Difficulty in Processing MSC

To evaluate task difficulty, we adopt *perplexity* as the primary metric, as it reflects a model's uncertainty when generating outputs. We measure the perplexity of various LLMs on MSC tasks. Specifically, we provide each model with dialogues ranging from one to four sessions and examine how task difficulty changes with longer dialogue contexts. In addition, we compare the perplexity of English and Korean MSC tasks to investigate whether LLMs struggle more with multilingual MSC than with English. Finally, we compare MSC tasks with other inputs of similar length to confirm that MSC is inherently more challenging than general long-context processing. Table 4 presents the perplexity results on the English and Korean MSC datasets. We observe two key findings. First, as shown in Table 4, all LLMs exhibit higher perplexity on the Korean MSC dataset compared to the English one, suggesting that processing Korean MSC is more challenging for LLMs. Second, the perplexity gap between the first and second sessions is relatively small compared to other session gaps. This is because the second-session dialogues typically refer to the content of the first session, providing related context that reduces difficulty. In contrast, the third and fourth sessions tend to diverge from earlier content, resulting in larger perplexity differences. Although the results in Table 4 suggest that Korean MSC is more challenging, an additional factor should be considered. In the context of LLM vocabularies, the tokenization of Korean may be less fine-grained than that of English. As a result, LLMs tend to generate more tokens when producing Korean text compared to semantically equivalent English outputs. In other words, the higher perplexity observed in the Korean MSC dataset may stem from the linguistic characteristics of Korean rather than from the intrinsic difficulty of the MSC task itself. To address this, we also measure the perplexity of other Korean tasks with lengths similar to that of the Korean MSC task. Table 5 presents the perplexity results on the Korean summarization task. As shown in Table 5, although the fourth-session dialogues in the MSC dataset and the documents in the summarization dataset have similar lengths, LLMs exhibit significantly higher perplexity on the MSC dataset. We hypothesize that this is because the topics or contents of multi-session dialogues tend to shift as the conversation progresses, whereas the topic of each document in the summarization dataset remains relatively consistent. This result indicates that the higher perplexity observed in

the Korean MSC dataset stems from the intrinsic difficulty of the task itself, rather than from the linguistic characteristics of Korean.

Table 4. Perplexity results on English and Korean MSC datasets. Each LLM is instructed to generate the final turn of every session, and perplexity is measured by averaging the token-level perplexities across the generated outputs.

Dataset	Model	Perplexity ↓ (Lower Is Better)			
		Session 1	Session 1–2	Session 1–3	Session 1–4
KEEM (KOR MSC)	EXAONE 4 32B	4.1	4.3	5.9	8.1
	Gemma 3 12B	3.9	4.2	5.4	7.5
	Llama 3.1 8B	3.5	4.2	5.3	7.8
	Qwen 2.5 7B	3.2	3.7	5.1	7.3
	Qwen 2.5 32B	2.8	3.1	5.0	7.1
	Qwen 3 8B	2.9	3.0	4.6	7.1
	Qwen 3 32B	2.4	2.4	4.2	6.7
MSC (ENG) [2]	EXAONE 4 32B	2.8	3.3	5.0	6.7
	Gemma 3 12B	3.1	3.1	3.8	5.4
	Llama 3.1 8B	2.6	2.7	3.9	5.6
	Qwen 2.5 7B	2.4	2.5	3.7	6.1
	Qwen 2.5 32B	2.1	1.8	3.2	5.3
	Qwen 3 8B	2.3	2.1	3.7	5.4
	Qwen 3 32B	1.9	1.7	3.1	4.9

Table 5. Perplexity results on Korean summarization datasets. The documents were sampled to have a similar length to the fourth-session dialogues in Table 4.

Task	Model	Perplexity
Summarization (KOR)	EXAONE 4 32B	5.3
	Gemma 3 12B	4.5
	Llama 3.1 8B	4.6
	Qwen 2.5 7B	4.4
	Qwen 2.5 32B	4.0
	Qwen 3 8B	4.4
	Qwen 3 32B	4.1

5.5. Results of Session Summarization

For the session summarization task, our objective is to accurately and separately generate the persona and episode memories. To this end, we fine-tune LLMs on multi-session dialogues using various training methods. We evaluate the quality of the generated summaries in terms of semantic similarity and informativeness. To compute these metrics, we use the GPT-5 API. For the semantic similarity metric, GPT-5 is instructed to assign a similarity score between 0 and 1.0 based on the semantic alignment between the generated summary and the gold summary in the test set. For the informativeness metric, GPT-5 is asked to rate the informativeness of the generated summary, given the corresponding session dialogue, on a scale from 1 to 5. Table 6 shows the performance results on the session summarization task across various methods. As shown in Table 6, all LLMs demonstrate strong performance on this task, indicating that session summarization is not particularly challenging for them. Qwen 3 8B achieves the best performance, even surpassing the EXAONE 4 32B model, which may possess a larger and more fine-grained Korean vocabulary. We hypothesize that the superior performance of Qwen 3 8B stems from its distillation from a larger model [32], allowing it to inherit the capabilities of the original model. In addition, we observe that LoRA and MoE show limited effectiveness compared to

the DPO and Neuron tuning methods. We attribute this phenomenon to the fact that LoRA and MoE do not modify the original model weights, whereas DPO and Neuron tuning directly update them. Compared to DPO, the neuron tuning method exhibits slightly better performance. This suggests that tuning a selected subset of important neurons can be more effective and efficient than adjusting all model weights.

Table 6. Performance results on the session summarization task. Each experiment was run five times, and the reported scores are the averages over these runs. Statistical significance was verified using a paired *t*-test; all pairwise comparisons between Neuron Tuning and other methods yield $p < 0.05$. Bold indicates the best result for each metric.

Model	Method	Semantic Similarity	Informativeness
EXAONE 4 32B	LoRA	0.87	4.48
	DPO	0.92	4.77
	MoE	0.86	4.59
	Layer Tuning	0.88	4.75
	Neuron Tuning	0.93	4.80
Gemma 3 12B	LoRA	0.88	4.51
	DPO	0.92	4.67
	MoE	0.88	4.60
	CPT	0.89	4.69
	Layer Tuning	0.89	4.73
	Neuron Tuning	0.94	4.82
Llama 3.1 8B	LoRA	0.85	4.35
	DPO	0.88	4.42
	MoE	0.85	4.39
	CPT	0.87	4.42
	Layer Tuning	0.86	4.50
	Neuron Tuning	0.90	4.64
Qwen 2.5 7B	LoRA	0.83	4.30
	DPO	0.89	4.44
	MoE	0.81	4.33
	CPT	0.90	4.49
	Layer Tuning	0.89	4.51
	Neuron Tuning	0.90	4.61
Qwen 3 8B	LoRA	0.89	4.72
	DPO	0.94	4.85
	MoE	0.91	4.74
	CPT	0.88	4.69
	Layer Tuning	0.93	4.81
	Neuron Tuning	0.95	4.92

5.6. Results of Memory Update

It is important to note that even in long-term conversations, memory updates are not always required after every session [3]. Therefore, to accurately evaluate the effectiveness of each method on the memory update task, we sample 500 multi-session dialogues in which memory updates are necessary (i.e., cases where the user's state changes in the subsequent session) in the test set of the KEEM. Following our previous work [3], we adopt *informativeness* and *conflict* as evaluation metrics to assess the quality of the memories generated by LLMs under each tuning method. To compute these metrics, we use the GPT-5 API. For informativeness, GPT-5 is instructed to evaluate how informative the generated memory is with respect to the entire session dialogue. For conflict, GPT-5 is asked to determine whether any inconsistency exists between the generated memory and

the dialogue, thereby verifying whether the memory update was performed accurately. Table 7 presents the performance results for the memory update task. Overall, across all methods and LLMs, the *informativeness* scores for the memory update task are lower than those for the session summarization task. This suggests that the memory update task is inherently more challenging. This is expected, as the session summarization task primarily requires compressing the content of a given session dialogue, whereas the memory update task involves integrating information from both the previous memory and the current dialogue to generate a newly updated memory. Moreover, models optimized for reasoning exhibit better performance on the memory update task. This suggests that reasoning capabilities are beneficial for memory updating. Next, the neuron tuning method achieves the best performance, consistent with its results on the session summarization task, demonstrating its overall effectiveness. In particular, we observe that neuron tuning is significantly more effective on the EXAONE model than on the other models. We attribute this to the characteristics of EXAONE: it was trained on only three languages—English, Korean, and Spanish—whereas the other models were trained on a wider variety of languages. Consequently, this factor may make it easier to identify Korean-specific neurons within EXAONE.

Table 7. Performance results on the memory update task. Each experiment was run five times, and the reported scores are the averages over these runs. Statistical significance was verified using a paired *t*-test; all pairwise comparisons between Neuron Tuning and other methods yield $p < 0.05$. Bold indicates the best result for each metric.

Model	Method	Informativeness	Conflict ↓
EXAONE 4 32B	LoRA	4.22	15.9%
	MoE	4.28	14.6%
	Layer Tuning	4.49	13.1%
	Neuron Tuning	4.67	9.8%
Gemma 3 12B	LoRA	4.27	14.5%
	MoE	4.31	14.4%
	CPT	4.40	13.3%
	Layer Tuning	4.32	14.6%
	Neuron Tuning	4.42	12.7%
Llama 3.1 8B	LoRA	4.14	17.3%
	MoE	4.12	17.0%
	CPT	4.25	14.9%
	Layer Tuning	4.25	14.5%
	Neuron Tuning	4.39	14.2%
Qwen 2.5 7B	LoRA	4.11	16.8%
	MoE	4.14	16.4%
	CPT	4.23	14.4%
	Layer Tuning	4.27	14.4%
	Neuron Tuning	4.40	14.0%
Qwen 3 8B	LoRA	4.54	13.6%
	MoE	4.52	13.7%
	CPT	4.61	11.5%
	Layer Tuning	4.60	11.1%
	Neuron Tuning	4.65	10.4%

To ensure more reliable assessment beyond automatic metrics, we conduct human evaluation. Five evaluators are instructed to assess the *informativeness* of generated memories and to identify any conflicts between the generated memory and the corresponding dialogue. We use the same sample set as in the automatic evaluation. To ensure feasibil-

ity, we limit the evaluation to the top three models ranked by GPT-5 scores. To prevent evaluator bias toward specific models or methods, we pool all generated memories from the three models and present them to evaluators in randomized order without revealing model identities. Table 8 presents the human evaluation results for the memory update task. Consistent with the automatic evaluation results, neuron tuning demonstrates significantly higher effectiveness compared to other methods.

Table 8. Human evaluation results on the memory update task. Inter-annotator agreement is assessed using Krippendorff’s Alpha with ordinal metric for *informativeness* ($\alpha = 0.82$, reliable) and Fleiss’ Kappa for *conflict* ($\kappa = 0.91$), both indicating substantial agreement. Bold indicates the best result for each metric.

Model	Method	Informativeness	Conflict ↓
EXAONE 4 32B	LoRA	3.67	17.5%
	MoE	3.83	16.4%
	Layer Tuning	3.88	14.7%
	Neuron Tuning	4.06	11.9%
Gemma 3 12B	LoRA	3.42	15.4%
	MoE	3.79	14.9%
	CPT	3.84	13.3%
	Layer Tuning	3.83	14.2%
	Neuron Tuning	3.94	12.1%
Qwen 3 8B	LoRA	3.35	15.6%
	MoE	3.65	15.7%
	CPT	3.86	13.5%
	Layer Tuning	3.82	14.3%
	Neuron Tuning	3.89	12.5%

5.7. Results of Response Generation

To accurately evaluate the response generation capability under the MSC setting, we use fifth-session dialogues designed in our previous work [3]. Each dialogue includes at least one turn discussing updated information, ensuring that the conversation explicitly engages with the updated memories. It is essential to note that we fine-tune LLMs using the updated memory M^U and the current session dialogue D_{cur} to generate responses, rather than relying on the accumulated session summarization S or the entire session dialogue D . This decision follows previous studies [2,3], which have demonstrated that our chosen approach is the most effective. We adopt two metrics to assess the quality of responses generated by the tuned models. The first is *engagement*, which measures how effectively the generated response engages with the user’s conversation, on a scale from 1 to 5. The second is *appropriateness of memory usage (AMU)*, which evaluates whether the model refers to the correct memory when generating its response. To measure these metrics, we use the GPT-5 API. Table 9 presents the performance results for the response generation task. Similar to the memory update task, reasoning capabilities are beneficial for generating responses that appropriately reference related memories. To produce an engaging response under the MSC setting, the model must reason about which memories to use and when to use them. Moreover, the neuron tuning method achieves the best performance across all LLMs, demonstrating its effectiveness. These results indicate that neuron tuning is more effective than LoRA, DPO, and MoE for fine-tuning LLMs on MSC tasks. However, there remains room for improvement in response generation.

Table 9. Performance results on the response generation task. Each experiment was run five times, and the reported scores are the averages over these runs. Statistical significance was verified using a paired *t*-test; all pairwise comparisons between Neuron Tuning and other methods yield $p < 0.05$. Bold indicates the best result for each metric.

Model	Method	Engagement	AMU ↑
EXAONE 4 32B	LoRA	3.78	85.9%
	MoE	3.73	84.7%
	Layer Tuning	3.99	84.6%
	Neuron Tuning	4.06	86.9%
Gemma 3 12B	LoRA	3.64	82.3%
	MoE	3.63	82.2%
	CPT	3.90	83.5%
	Layer Tuning	3.91	83.4%
	Neuron Tuning	3.98	83.9%
Llama 3.1 8B	LoRA	3.62	78.9%
	MoE	3.66	78.4%
	CPT	3.84	81.9%
	Layer Tuning	3.82	81.6%
	Neuron Tuning	3.89	82.7%
Qwen 2.5 7B	LoRA	3.59	77.4%
	MoE	3.61	77.3%
	CPT	3.79	81.6%
	Layer Tuning	3.72	80.4%
	Neuron Tuning	3.87	81.7%
Qwen 3 8B	LoRA	3.73	85.0%
	MoE	3.78	85.0%
	CPT	3.88	85.8%
	Layer Tuning	3.82	85.4%
	Neuron Tuning	3.97	87.4%

To ensure more reliable assessment beyond automatic metrics, we conduct human evaluation. For the response generation task, we introduce an additional metric, *naturalness*, which evaluates the fluency and naturalness of generated responses in Korean on a 5-point Likert scale. We exclude this metric from automatic evaluation because even GPT-5 is unreliable for assessing *naturalness* in Korean. The correlation between automatic and human evaluation scores on *naturalness* ($\rho = 0.58$) is substantially lower than that of other metrics, indicating that current frontier models struggle to adequately assess Korean naturalness compared to other evaluation criteria. As shown in Table 10, consistent with the automatic evaluation results, neuron tuning outperforms all baselines across every metric, demonstrating its effectiveness for response generation in MSC settings. Notably, Qwen3 8B exhibits relatively lower performance compared to EXAONE4 32B and Gemma3 12B. Further analysis reveals that Qwen3 8B frequently generates Chinese characters within its responses, resulting in low scores from human evaluators.

Table 10. Human evaluation results on the response generation task. Inter-annotator agreement is assessed using Krippendorff’s Alpha with ordinal metric for *engagement* and *naturalness* ($\alpha_e = 0.83$, $\alpha_n = 0.87$) and Fleiss’ Kappa for *AMU* ($\kappa_c = 0.90$), both indicating substantial agreement. Bold indicates the best result for each metric.

Model	Method	Engagement	Naturalness	AMU ↑
EXAONE 4 32B	LoRA	3.57	4.48	83.1%
	MoE	3.57	4.40	82.4%
	Layer Tuning	3.69	4.51	81.3%
	Neuron Tuning	3.90	4.54	83.5%
Gemma 3 12B	LoRA	3.44	4.32	80.6%
	MoE	3.41	4.33	80.2%
	CPT	3.77	4.46	81.8%
	Layer Tuning	3.72	4.45	81.6%
	Neuron Tuning	3.85	4.53	83.0%
Qwen 3 8B	LoRA	3.37	4.10	79.2%
	MoE	3.40	4.12	80.0%
	CPT	3.69	4.18	80.4%
	Layer Tuning	3.65	4.19	80.5%
	Neuron Tuning	3.78	4.24	81.6%

5.8. Effectiveness of Dividing Session Summarization Type

Table 11 presents the perplexity and engagement scores when persona and episode memory are considered separately and vice versa. As Table 11 demonstrates, the division of persona and episode memory results in slightly improved performance for generating responses in long-term conversation scenarios. This suggests that distinguishing between long-term information (such as user introductions, personal details, preferences, possessions, relationships, and aspirations) and short-term episodic information can enhance the ability of the model to generate more engaging responses.

Table 11. Results of automatic evaluation on the generated response according to dividing persona and episode memory. The scores were averaged across all LLMs. Bold indicates the best result for each metric.

Method	Session 2		Session 3		Session 4	
	PPL ↓	Eng. ↑	PPL ↓	Eng. ↑	PPL ↓	Eng. ↑
w/o divide	3.9	3.94	5.4	3.69	6.7	3.53
w/ divide	3.1	4.32	4.0	4.11	5.1	3.92

6. Analysis

6.1. Effectiveness of Multi-Task Learning

So far, we have fine-tuned LLMs on each task independently. However, since the three tasks are all related to conducting MSC, we additionally fine-tune LLMs under multi-task and continual learning settings to investigate which approach is more effective in these scenarios. For multi-task learning, we fine-tune LLMs on all three tasks simultaneously. For continual learning, we fine-tune LLMs on the session summarization, memory update, and response generation tasks sequentially. Figure 2 presents the results of multi-task and continual learning across the three tasks. As shown in the figure, multi-task learning improves performance on certain tasks, implying that jointly learning multiple tasks may facilitate beneficial interactions among them. In contrast, continual learning substantially degrades the performance of all tasks except for the neuron tuning method. The LoRA, CPT, and Layer tuning methods lose the ability to perform previously learned tasks while

acquiring new capabilities, suggesting that these approaches are not well-suited for continual learning. We speculate that the limited capacity of LoRA restricts its ability to retain knowledge from multiple tasks. In the case of CPT, although it updates all model parameters, it still suffers from catastrophic forgetting. By contrast, the neuron tuning method preserves prior task performance, as it modifies only a small subset of weights responsible for language-specific capabilities, such as Korean. This finding underscores the effectiveness of neuron tuning for continual learning.



Figure 2. Results of multi-task and continual learning across all methods.

6.2. Comparative Evaluation of Neuron Identifying Methodologies

To demonstrate the effectiveness of our neuron identification methodology, we fine-tune the neurons detected by our method and by LAPE, respectively, and then compare their results. As mentioned earlier, we first analyze the neuron distribution from our method across all layers according to whether activation score normalization is applied. As shown in Figure 3, without normalization, neurons are concentrated in specific layers (Figure 3a,c). In contrast, with normalization, neurons are more evenly distributed across layers (Figure 3b,d). Because the distribution of neurons differs depending on whether normalization is applied, the effect of neuron tuning may also vary. Intuitively, we can expect that tuning neurons identified without normalization would resemble layer-specific tuning, since those neurons are highly concentrated within certain layers. Figure 4 presents

the performance results of neuron tuning according to the neuron detection method. Tuning neurons identified without normalization significantly degrades the performance of both the memory update and response generation tasks. This finding is consistent with the observations of [9], which reported that LLMs tend to transfer multilingual inputs into English representations in shallow layers, perform reasoning in English in the middle layers, and generate outputs in the original language in the top layers. Because neurons identified without normalization do not exist in the shallow layers, tuning these neurons is not beneficial for improving LLMs' understanding of Korean inputs. Moreover, tuning neurons identified by the LAPE method also leads to decreased performance on both tasks. Since LAPE focuses solely on entropy across languages, the neurons it identifies are typically associated with generating Korean outputs rather than understanding Korean inputs or supporting reasoning. These results demonstrate that our neuron identification method—which jointly considers both activation scores and entropy—is effective in detecting critical neurons that contribute to understanding Korean inputs within English representations and reasoning processes.

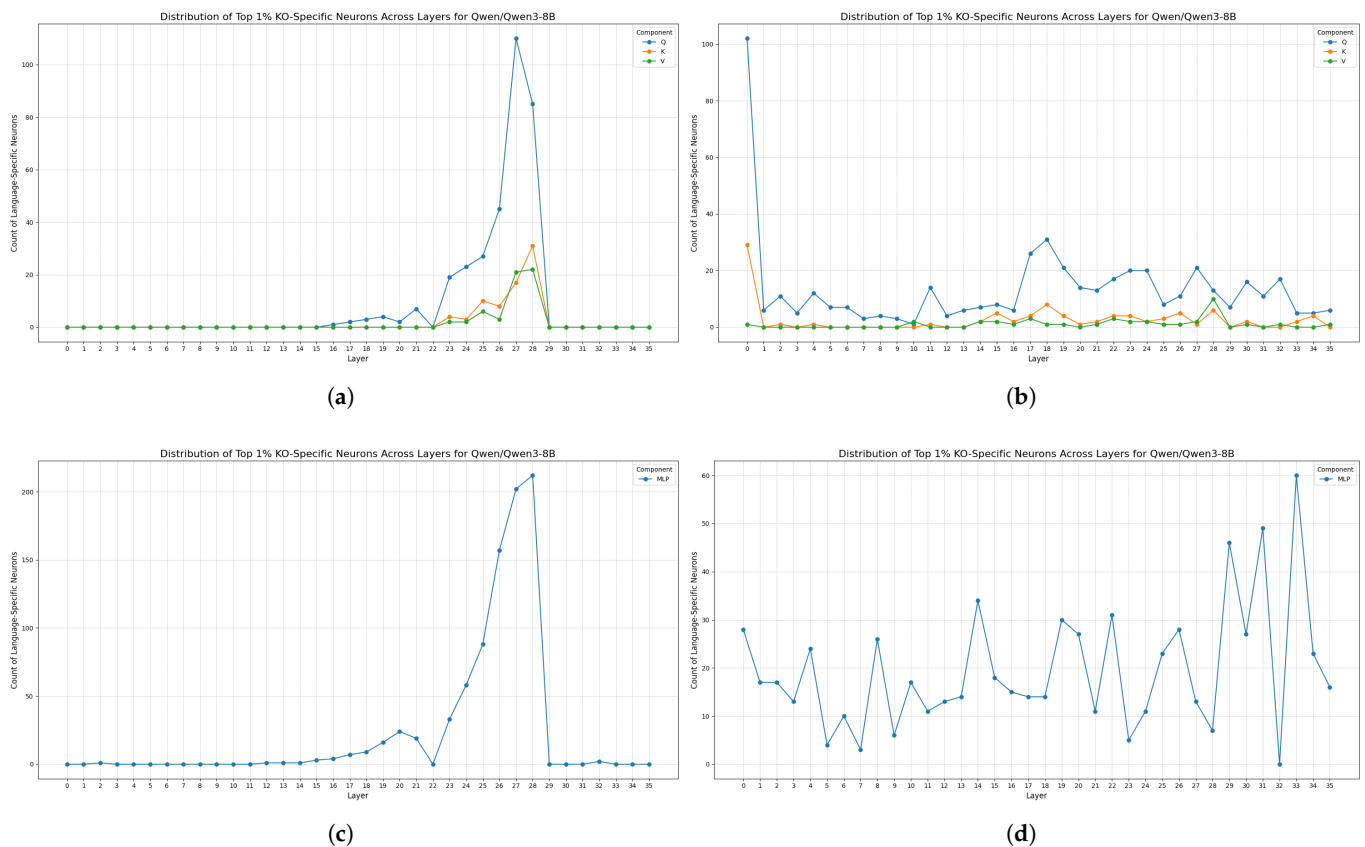


Figure 3. Distribution of Korean-specific neurons identified by our method across all layers, with and without activation score normalization. (a) Distribution of Qwen3-8B attention module's neuron w/o normalization. (b) Distribution of Qwen3-8B attention module's neuron w/ normalization. (c) Distribution of Qwen3-8B MLP module's neuron w/o normalization. (d) Distribution of Qwen3-8B MLP module's neuron w/ normalization.

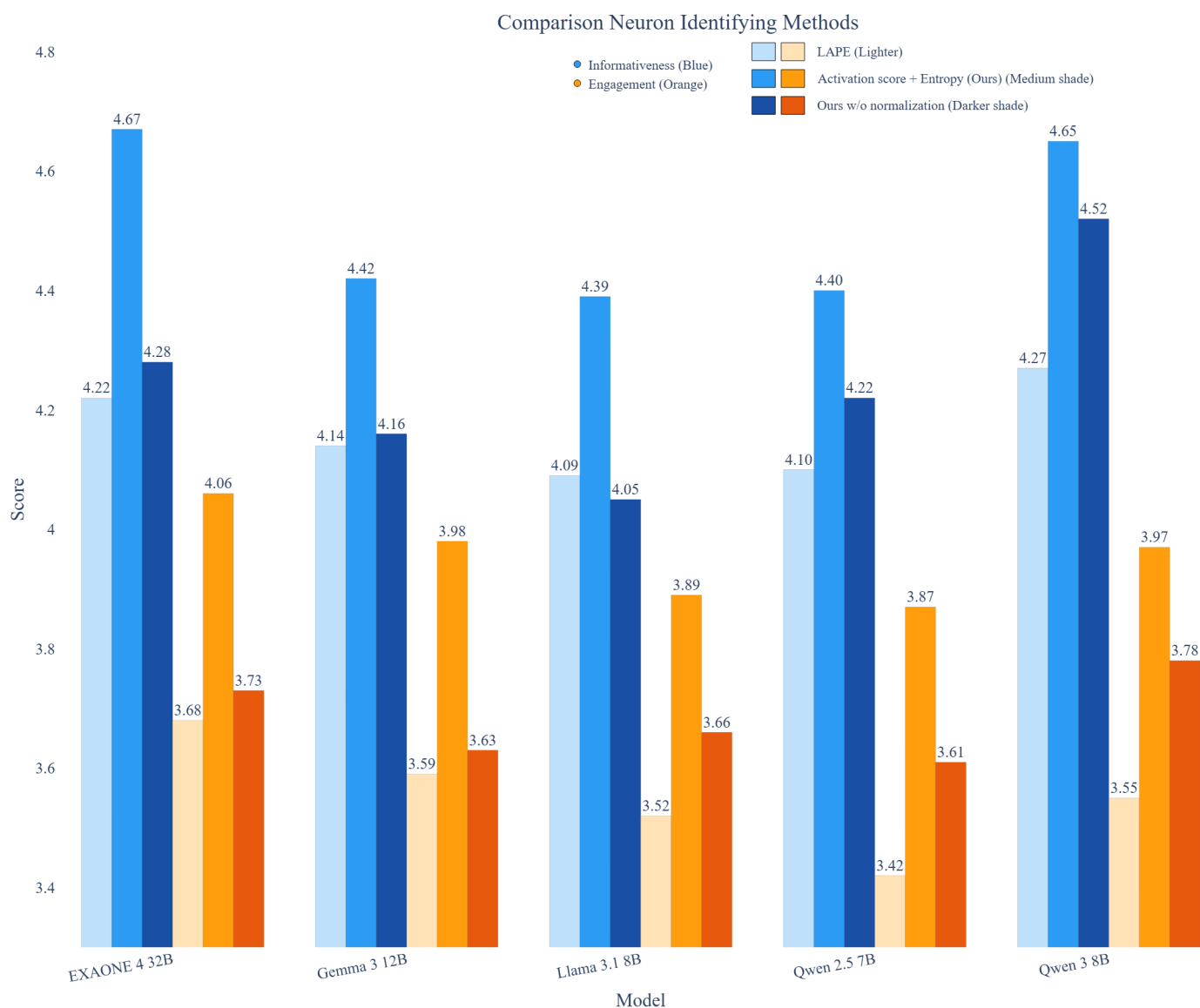


Figure 4. Performance results on the memory update (Informativeness) and response generation (Engagement) tasks across neuron identifying methods. Each experiment was run three times, and the reported scores are the averages over these runs.

7. Conclusions

In this study, we conducted a comprehensive empirical analysis of LLMs for long-term conversations (i.e., MSC) in Korean. Our investigation revealed that Korean MSC poses significantly greater challenges for current LLMs than its English counterpart, highlighting the limitations of multilingual language understanding in long-term conversational contexts. To address this issue, we explored multiple training paradigms—LoRA, DPO, MoE, and Neuron Tuning—across key tasks in memory-augmented dialogue: session summarization, memory update, and response generation. Experimental results demonstrated that Neuron Tuning consistently outperforms other methods, achieving both effectiveness and efficiency by selectively fine-tuning neurons responsible for language-specific and reasoning capabilities. Furthermore, we proposed an improved neuron identification method that integrates activation scores and entropy, overcoming the limitations of existing approaches such as LAPE. The identified neurons not only yielded superior task performance but also provided deeper insight into the internal mechanisms of multilingual reasoning. In addition, our analysis on multi-task and continual learning revealed that while multi-task learning

can foster positive task interaction, most tuning methods suffer from catastrophic forgetting under continual learning—except Neuron Tuning, which preserves prior knowledge due to its localized adaptation strategy. These findings underscore the potential of neuron-level tuning as a scalable and interpretable approach for improving LLMs' multilingual long-term conversational ability. In future work, we plan to extend our study to other underrepresented languages and explore neuron-level cross-lingual transfer for continual learning. We also aim to integrate explicit reasoning modules and dynamic memory retrieval mechanisms to further enhance the coherence and adaptability of long-term dialogue in multilingual settings.

Author Contributions: Conceptualization, H.K. (Hongjin Kim), J.K. and H.K. (Harksoo Kim); Methodology, H.K. (Hongjin Kim), J.K. and H.K. (Harksoo Kim); Validation, H.K. (Hongjin Kim), J.K., Y.J. and Y.S.; Formal analysis, H.K. (Hongjin Kim), J.K., Y.J., Y.S. and H.K. (Harksoo Kim); Investigation, H.K. (Hongjin Kim), J.K., Y.J., Y.S. and H.K. (Harksoo Kim); Resources, H.K. (Hongjin Kim) and J.K.; Data curation, H.K. (Hongjin Kim) and J.K.; Writing—original draft preparation, H.K. (Hongjin Kim) and J.K.; Writing—review and editing, H.K. (Hongjin Kim), J.K., Y.J., Y.S. and H.K. (Harksoo Kim); Visualization, H.K. (Hongjin Kim), J.K., Y.J. and Y.S.; Supervision, H.K. (Harksoo Kim); Project administration, H.K. (Harksoo Kim); funding acquisition, H.K. (Harksoo Kim). All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by Konkuk University Researcher Fund in 2024. This work was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the virtual convergence support program to nurture the best talents (IITP-2026-RS-2023-00256615) grant funded by the Korea government (MSIT).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the authors due to licensing constraints associated with the original source data.

Acknowledgments: We thank the members of the NLP laboratory at Konkuk University for their technical support.

Conflicts of Interest: Author Yujin Sim is currently employed by the company SweetK. However, this research was conducted while the author was affiliated with Konkuk University prior to joining the company. SweetK was not involved in the study design, data collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication. The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LLM	Large Language Model
MSC	Multi-Session Conversation
PLM	Pre-trained Language Model
LoRA	Low-Rank Adaptation
DPO	Direct Preference Optimization
MoE	Mixture-of-Experts
RAG	Retrieval-Augmented Generation
KEEM	Keep Emotion and Essential Memory
LAPE	Language Activation Probability Entropy
PLND	Parallel Language-specific Neuron Detection
CPT	Continual Pre-Training
RL	Reinforcement Learning
RLHF	Reinforcement Learning with Human Feedback
AMU	Appropriateness of Memory Usage

Appendix A

Details of Instructions

Tables A1 and A2 present the instructions for persona–episode memory separation and memory update, respectively.

Table A1. Instruction for separating persona and episode memory.

Instruction (Translated into English)
Given a dialogue between a user and a chatbot, along with its summarized memory, separate the memory into persona memory and episode memory.
Persona memory should include long-term information about the user—details that are unlikely to change frequently. Examples include the user’s name, place of residence, occupation, preferences, personality traits, and other stable personal attributes.
Episode memory should include short-term information that often changes based on the user’s recent experiences or events. Examples include recent important events in the user’s life, current emotions and their causes, and other context-specific or time-sensitive details.

Table A2. Instruction for memory update.

In cases where the information in [speaker’s summary] requires modifying or deleting a sentence in [Persona Memory] or [Episode Memory], you must update the corresponding sentence to reflect the most recent information while maintaining overall consistency.
Persona Memory contains long-term user information (e.g., name, residence, occupation, stable preferences) accumulated across past conversations. Episode Memory contains short-term or changeable information (e.g., recent events, current emotions and their causes).
Because [speaker’s summary] reflects the current conversation, it always contains the most recent information. Therefore, when discrepancies arise, you must update [Persona Memory] or [Episode Memory] using the content of [speaker’s summary].
You must not merge, modify, or delete sentences in Persona Memory or Episode Memory solely because they address the same topic. You should update a sentence only when it must be changed based on the new information in [speaker’s summary], or when the information naturally continues.
When updating, ensure that no new content is invented and that no existing content is lost.

References

1. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7871–7880.

2. Xu, J.; Szlam, A.; Weston, J. Beyond Goldfish Memory: Long-Term Open-Domain Conversation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 5180–5197.

3. Kang, J.; Kim, H.; Kim, H. Generation-Based and Emotion-Reflected Memory Update: Creating the KEEM Dataset for Better Long-Term Conversation. In Proceedings of the 31st International Conference on Computational Linguistics, Abu Dhabi, United Arab Emirates, 19–24 January 2025; pp. 9260–9277.
4. Kim, H.; Keum, B.; Huang, J.; Kwon, O.; Kim, H. Multi-Level Attention-Based Generation Model for Long-Term Conversation. *J. KIISE* **2025**, *52*, 117–124. [[CrossRef](#)]
5. Ko, H.; Son, G.; Choi, D. Understand, Solve and Translate: Bridging the Multilingual Mathematical Reasoning Gap. *arXiv* **2025**, arXiv:2501.02448. [[CrossRef](#)]
6. Han, S.; Suk, J.; An, S.; Kim, H.; Kim, K.; Yang, W.; Choi, S.; Shin, J. Trillion 7B Technical Report. *arXiv* **2025**, arXiv:2504.15431. [[CrossRef](#)]
7. Yoo, H.; Park, C.; Yun, S.; Oh, A.; Lee, H. Code-Switching Curriculum Learning for Multilingual Transfer in LLMs. *arXiv* **2024**, arXiv:2411.02460. [[CrossRef](#)]
8. Laban, P.; Hayashi, H.; Zhou, Y.; Neville, J. LLMs Get Lost in Multi-Turn Conversation. *arXiv* **2025**, arXiv:2505.06120. [[CrossRef](#)]
9. Zhao, Y.; Zhang, W.; Chen, G.; Kawaguchi, K.; Bing, L. How Do Large Language Models Handle Multilingualism? In Proceedings of the Thirty-Eighth Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 9–15 December 2024.
10. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2022**, arXiv:2106.09685.
11. Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C.D.; Ermon, S.; Finn, C. Direct Preference Optimization: Your Language Model Is Secretly a Reward Model. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 53728–53741.
12. Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; Chi, E.H. Modeling Task Relationships in Multi-Task Learning with Multi-Gate Mixture-of-Experts. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1930–1939.
13. Kim, H.; Lee, J.; Lee, K.; Shin, J.; Lim, S.; Kwon, O. Do LLMs Need Inherent Reasoning Before Reinforcement Learning? A Study in Korean Self-Correction. In Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, Mumbai, India, 20–24 December 2025; pp. 527–542.
14. Wang, M.; Zhang, N.; Xu, Z.; Xi, Z.; Deng, S.; Yao, Y.; Zhang, Q.; Yang, L.; Wang, J.; Chen, H. Detoxifying large language models via knowledge editing. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 11–16 August 2024; pp. 3093–3118.
15. Tang, T.; Luo, W.; Huang, H.; Zhang, D.; Wang, X.; Zhao, W.X.; Wei, F.; Wen, J.-R. Language-Specific Neurons: The Key to Multilingual Capabilities in Large Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 11–16 August 2024; pp. 5701–5715.
16. Xu, H.; Zhan, R.; Ma, Y.; Wong, D.F.; Chao, L.S. Let's Focus on Neuron: Neuron-Level Supervised Fine-tuning for Large Language Model. In Proceedings of the 31st International Conference on Computational Linguistics, Abu Dhabi, United Arab Emirates, 19–24 January 2025; pp. 9393–9406.
17. Bae, S.; Kwak, D.; Kang, S.; Lee, M.Y.; Kim, S.; Jeong, Y.; Kim, H.; Kim, H.; Lee, S.-W.; Park, W.; et al. Keep Me Updated! Memory Management in Long-term Conversations. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 3769–3787.
18. Jia, Z.; Liu, Q.; Li, H.; Chen, Y.; Liu, J. Evaluating the long-term memory of large language models. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, Vienna, Austria, 27 July–1 August 2025; pp. 19759–19777.
19. Tan, Z.; Yan, J.; Hsu, I.H.; Han, R.; Wang, Z.; Le, L.; Song, Y.; Chen, Y.; Palangi, H.; Lee, G.; et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 27 July–1 August 2025; pp. 8416–8439.
20. Kim, E.; Park, C.; Chang, B. SHARE: Shared Memory-Aware Open-Domain Long-Term Dialogue Dataset Constructed from Movie Script. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 27 July–1 August 2025; pp. 14474–14498.
21. OpenAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774. [[CrossRef](#)]
22. Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N.A.; Khashabi, D.; Hajishirzi, H. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 13484–13508.
23. Cegin, J.; Simko, J.; Brusilovsky, P. ChatGPT to Replace Crowdsourcing of Paraphrases for Intent Classification: Higher Diversity and Comparable Model Robustness. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 1889–1905.
24. Chen, Z.; Shen, Y.; Ding, M.; Chen, Z.; Zhao, H.; Learned-Miller, E.G.; Gan, C. Mod-Squad: Designing Mixtures of Experts as Modular Multi-Task Learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023, Vancouver, BC, Canada, 17–24 June 2023; pp. 11828–11837.

25. Dai, D.; Deng, C.; Zhao, C.; Xu, R.; Gao, H.; Chen, D.; Li, J.; Zeng, W.; Yu, X.; Wu, Y.; et al. DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 11–16 August 2024.
26. Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. DeepSeek-V3 Technical Report. *arXiv* **2024**, arXiv:2412.19437.
27. Liu, W.; Xu, Y.; Xu, H.; Chen, J.; Hu, X.; Wu, J. Unraveling Babel: Exploring Multilingual Activation Patterns of LLMs and Their Applications. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; pp. 11855–11881.
28. LG AI Research; Bae, K.; Choi, E.; Choi, K.; Choi, S.J.; Choi, Y.; Han, K.; Hong, S.; Hwang, J.; Hwang, T.; et al. EXAONE 4.0: Unified Large Language Models Integrating Non-reasoning and Reasoning Modes. *arXiv* **2025**, arXiv:2507.11407. [[CrossRef](#)]
29. Gemma Team; Kamath, A.; Ferret, J.; Pathak, S.; Vieillard, N.; Merhej, R.; Perrin, S.; Matejovicova, T.; Ramé, A.; Rivière, M.; et al. Gemma 3 Technical Report. *arXiv* **2025**, arXiv:2503.19786. [[CrossRef](#)]
30. AI@Meta. Llama 3 Model Card. 2024. Available online: <https://github.com/meta-llama/llama3> (accessed on 1 January 2025).
31. Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. Qwen Technical Report. *arXiv* **2023**, arXiv:2309.16609. [[CrossRef](#)]
32. Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. Qwen3 Technical Report. *arXiv* **2025**, arXiv:2505.09388. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.