⏳ Ticktack : Long Span Temporal Alignment of Large Language Models Leveraging Sexagenary Cycle Time Expression

Anonymous ACL submission

Abstract

Large language models (LLMs) suffer from temporal misalignment issues especially across long span of time. The issue stems from knowing that LLMs are trained on vast amounts of data with sparse temporal information over long periods, such as thousands of years, resulting in insufficient learning or catastrophic forgetting by the LLMs. This paper proposes a methodology named "Ticktack" for addressing the LLM's long-time span misalignment in a yearly setting. Specifically, we first propose to utilize the sexagenary year expression instead of the Gregorian year expression employed by LLMs, achieving a more uniform distribution in yearly granularity. Then, we employ polar coordinates to model the sexagenary cycle of 60 terms and the year order within each term, with additional temporal encoding to ensure LLMs understand them. Finally, we present a temporal representational alignment approach for post-training LLMs that effectively distinguishes time points with relevant knowledge, hence improving performance on time-related tasks, particularly over a long period. We also create a long time span benchmark for evaluation. Experimental results prove the effectiveness of our proposal.

1 Introduction

Language models have always suffered from temporal misalignment issues stemming from the temporal discrepancies between the training and testing data, resulting in variability in reference time during downstream tasks (Lazaridou et al., 2021; Luu et al., 2022; Jaidka et al., 2018; Tan et al., 2023). The issues persist with recently released large language models (LLMs) such as LLama (Touvron et al., 2023a) and GPT-4 (Achiam et al., 2023), which are trained on enormous datasets and exhibit significant performance decreases over time, especially when the time periods are long (Zhao et al., 2024; Nylund et al., 2023; Luu et al., 2022).

The long-span temporal misalignment issues in LLMs primarily arise from the extensive training data covering thousands of years (e.g., from BCE to post-2000 AD). The enormous training data generally lacks explicit temporal grounding, resulting in relatively limited and sparse knowledge of specific time periods (Drinkall et al., 2024). We investigate the distribution of years in the wiki dataset [1] and Baidu Baike[2], as illustrated in Figure 1. It indicates that data is rare in ancient ages, such as BCE, but concentrated in the internet era (1990s–present). Note that the year refers to the temporal reference of the data's content. Other studies (Yang et al., 2023) show similar findings. The sparse and long-tail distribution of training data over time results in insufficient learning or catastrophic forgetting in LLMs, leading to even poor performance during low-resource years (McCoy et al., 2023; Razeghi et al., 2022).

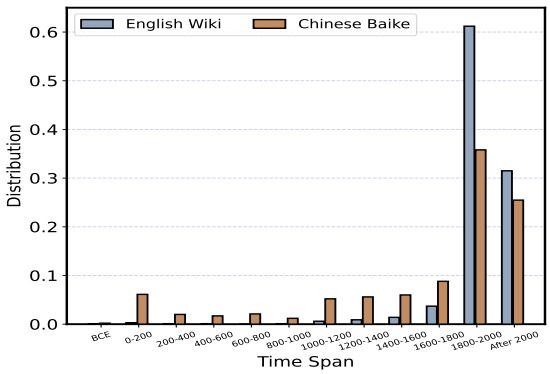


Figure 1: The distribution of temporal information in both Wikipedia (English) and Baidu Baike (Chinese), with statistics conducted at intervals of 200 years from BCE to after 2000.

Existing approaches (Mitchell et al., 2021; Meng et al., 2022) to resolving time misalignment issues emphasize updating models with new knowledge,

[1] https://huggingface.co/datasets/wikimedia/wikipedia

[2] https://baike.baidu.com/

yet they do not assess the internal temporal knowledge of LLMs over long periods. The most relevant work for us is Wei et al. (2025), which divides the time span from Pre-Qin to Modern into three distinct periods. This classification is too coarse for reasoning in LLMs.

This paper proposes a plug-and-play methodology named "Ticktack" for addressing the LLM's long-time span misalignment in a yearly setting. **To begin**, we propose solving the sparse and long-tail distribution of training data throughout time by employing a novel sexagenary year expression instead of the Gregorian year expression used by LLMs. The idea is founded on the observations of Tan et al. (2023) that the Gregorian year expression employed by LLMs resulted in an excessively wide range for the year embedding within the representation space of LLMs. Sexagenary time expression could achieve a more uniform distribution in yearly granularity. Figure 4 in Section 4.1 provides a comprehensive analysis. **Subsequently**, we apply polar coordinates to represent the sexagenary cycle of 60 terms and the chronological sequence inside each term, integrating additional temporal encoding to facilitate comprehension by LLMs. **Finally**, we present a temporal representational alignment approach for post-training LLMs that effectively distinguishes time points with relevant knowledge, thereby improving performance on time-related tasks, particularly over a long period.

Due to the lack of long time span benchmarks, we develop TempLS, a question-answering dataset covering the period from 75,000 BCE to 2025 AD, to facilitate the analysis of Ticktack's efficiency. We conduct experiments over several representative open-source LLMs ranging from 3 billion to 13 billion parameters. Experimental results on both open-source time-related benchmarks and TempLS prove the effectiveness of our proposal.

2 Related work

Temporal expressions and embeddings in language models. Traditional works (Yang et al., 2023) use a normalized value for time expression using tools such as SUTIME (Chang and Manning, 2012). With the development of pre-trained language models, researchers try to explore better time expression. In order to have a more comprehensive understanding of temporal expressions, Tan et al. (2023) divifgfdes 1900 to 2040 into seven 20-year time periods. Zhang and Choi (2023) leverages duration statistics on each dataset's development, such as seconds and minutes. However, all these time expressions within the scope of the Gregorian calendar system still suffer from the complicated representation space. Wei et al. (2025) segments the Chinese lexicon history into three periods: Ancient, Middle Ancient, and Near Ancient, and uses a one-hot embedding to represent them. This classification is too coarse for reasoning in LLMs. Recent LLMs, such as GPT-4 (Achiam et al., 2023), tokenize numeric information independently from a perspective of tokenization. However, this approach lost the distinct meaning of terms like "2014" as a specific year.

Temporal alignment of language models. Early efforts (Borkakoty and Espinosa-Anke, 2024; Dhingra et al., 2022) focus on designing novel datasets to probe LMs for temporal-related understanding, while more recent works (Wang and Zhao, 2023; Jang et al., 2022; Margatina et al., 2023) introduce new benchmarks to evaluate the temporal alignment capabilities. To create temporally adapted language models, conventional methods rely on continual learning on time-specific data (Agarwal and Nenkova, 2022; Loureiro et al., 2022). Lately, some knowledge modification techniques are proposed to align temporal knowledge (Zhu et al., 2020; De Cao et al., 2021; Dai et al., 2022). Other works (Zhao et al., 2024; Longpre et al., 2024; Gurnee and Tegmark, 2023) study the LMs' temporal misalignment caused by the chaotic pretraining corpus (Longpre et al., 2024) and LMs can represent temporal knowledge learned from pretraining in their internal states (Gurnee and Tegmark, 2023). These findings open up the possibility of aligning models to a specific time.

Jia Zi	Yi Chou	Bing Yin	Ding Mao	Wu Chen	Ji Si	Geng Wu	Xin Wei	Ren Shen	Gui You
1864	1865	1866	1867	1868	1869	1870	1871	1872	1873
1924	1925	1926	1927	1928	1929	1930	1931	1932	1933
1984	1985	1986	1987	1988	1989	1990	1991	1992	1993
Jia Xu	Yi Hai	Bing Zi	Ding Chou	Wu Yin	Ji Mao	Geng Chen	Xin Si	Ren Wu	Gui Wei
1874	1875	1876	1877	1878	1879	1880	1881	1882	1883
1934	1935	1936	1937	1938	1939	1940	1941	1942	1943
1994	1995	1996	1997	1998	1999	2000	2001	2002	2003
Jia Shen	Yi You	Bing Xu	Ding Hai	Wu Zi	Ji Chou	Geng Yin	Xin Mao	Ren Chen	Gui Si
1884	1885	1886	1887	1888	1889	1890	1891	1892	1893
1944	1945	1946	1947	1948	1949	1950	1951	1952	1953
2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
Jia Wu	Yi Wei	Bing Shen	Ding You	Wu Xu	Ji Hai	Geng Zi	Xin Chou	Ren Yin	Gui Mao
1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
1954	1955	1956	1957	1958	1959	1960	1961	1962	1963
2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
Jia Chen	Yi Si	Bing Wu	Ding Wei	Wu Shen	Ji You	Geng Xu	Xin Hai	Ren Zi	Gui Chou
1904	1905	1906	1907	1908	1909	1910	1911	1912	1913
1964	1965	1966	1967	1968	1969	1970	1971	1972	1973
2024	2025	2026	2027	2028	2029	2030	2031	2032	2033
Jia Yin	Yi Mao	Bing Chen	Ding Si	Wu Wu	Ji Wei	Geng Shen	Xin You	Ren Xu	Gui Hai
1914	1915	1916	1917	1918	1919	1920	1921	1922	1923
1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
2034	2035	2036	2037	2038	2039	2040	2041	2042	2043

Figure 2: The correspondence between the sexagenary year (blue) and Gregorian year (black). For instance, both 1864 and 1924 correspond to the "Jiazi" year.

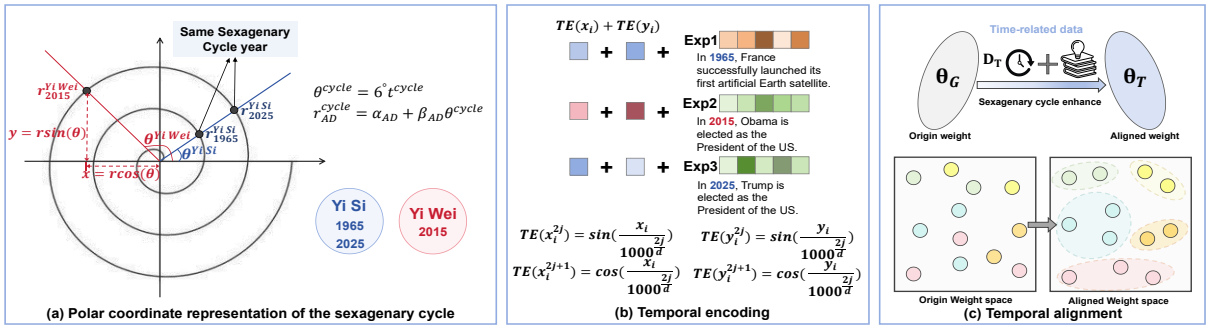


Figure 3: An overview of Ticktack (a) illustrates the novel way to express the years, leveraging the polar coordinate representation of the sexagenary cycle. (b) adopts sine and cosine functions to encode temporal information based on the sexagenary cycle. (c) describes the temporal alignment process to further transform the original weight space of the LLMs into a temporal re-organized and distinguished weight space.

3 Methodology

3.1 Preliminary: conversion between sexagenary year and Gregorian year

We initially provide the concept of the sexagenary cycle chronology and its correlation with the Gregorian calendar year, which is primarily utilized by LLMs during data preprocessing. The sexagenary cycle generates a sixty-year cycle utilized in the calendars of China and several other Far Eastern nations, derived from the combination of two fundamental cycles of ten and twelve[3]. Figure 2 depicts the sexagenary cycle chronology, which divides years into 60 categories, ranging from "Jiazi" to "Guihai." The associated Gregorian calendar years are grouped under their sexagenary category. For instance, the years 1864 AD and 1924 AD both correspond to the "Jiazi" year in the sexagenary cycle chronology.

By employing the sexagenary cycle chronology to represent the years, thousands of years of long-term data are reconstructed and aggregated into a 60-year cycle. As a result, the time representation achieves a more uniform distribution than the broader distribution space in the Gregorian year system, allowing for even better connections with relevant events. More detailed analyses are listed in Figure 4 of Section 4.1.

3.2 Overview of Ticktack

Ticktack is a plug-and-play methodology for LLMs that translates and aligns the year representation from the Gregorian calendar system with the sexagenary cycle chronology, hence enhancing the performance of LLMs on temporal tasks over long spans. Figure 3 illustrates the pipelines of Ticktack. It consists of three modules: (a) a polar coordinate representation of the sexagenary years; (b) temporal encoding; and (c) temporal alignment of weight space.

Firstly, to make the LLMs understand the newly introduced sexagenary year expression, we utilize the polar coordinate to represent the sexagenary cycle of 60 terms and the Gregorian years order within each of the 60 categories. Then, we design a temporal encoding method to inject the newly introduced sexagenary year information into its associated input data embedding. Finally, we defined a temporal alignment objective to post-train the LLMs using the time-encoded input data. In this way, the pre-trained weight space of the LLMs is transformed into a temporal re-organized and distinguished weight space, enhancing the comprehension of long-span temporal information with relevant knowledge. In the next subsections, we will present the specifics for each module.

3.3 Polar coordinate representation of the sexagenary year

By using polar coordinates to represent the sexagenary year, we aim to trace the continuity of time information over a cycle period and bridge the representation similarity between years with a 60-year interval based on sexagenary cycle chronology.

Given an input sequence s_i of length l, the input embedding of s_i generated by LLM is denoted as $h_i \in \mathbb{R}^{l \times d}$, where d represents the hidden dimension. We define t_i^{AD} to indicate the Gregorian year tokens in the input sequence s_i (e.g. "t_i^{AD} = 1965" in "France successfully launched its first artificial Earth satellite in 1965."). The Gregorian year t_i^{AD} could be transformed to the sexagenary year t_i^{cycle},

[3] https://en.wikipedia.org/wiki/Sexagenary_cycle

according to the following Eq. (1):

$$t_i^{cycle}: \begin{cases} (60 - |t_i^{AD}| - 2)\ mod\ 60, & t_i^{AD} < 0 \\ (60 - |t_i^{AD} - 3|)\ mod\ 60, & 0 < t_i^{AD} < 4 \\ (t_i^{AD} - 3)\ mod\ 60, & t_i^{AD} \geq 4 \end{cases} \tag{1}$$

To make the LLM understand the sexagenary year, we utilize the polar coordinate to represent t_i^{cycle}. As illustrated in Eq. (2), we use θ_{cycle} to identify the 60 terms or categories in a sexagenary cycle. r_{AD}^{cycle} is used to differentiate years within one category of a sexagenary cycle. That is to say, an ensemble of years within one category of a sexagenary cycle share the same angle but have different distances from the pole in polar coordinates. α_{AD} and β_{AD} are hyperparameters to determine the r_{AD}^{cycle}.

$$\begin{aligned} r_{AD}^{cycle} &= \alpha_{AD} + \beta_{AD}\theta_{cycle} \\ \theta_{cycle} &= \frac{360^\circ}{60} t_i^{cycle} = 6^\circ t_i^{cycle} \end{aligned} \tag{2}$$

As seen in Figure 3(a), 1965 AD and 2025 AD both belong to the "Yi Si" of the sexagenary year and hence share the same θ_{cycle} value. To distinguish them under "Yi Si," by adjusting the values of α_{AD} and β_{AD}, 2025 AD has a larger r_{2025}^{YiSi} compared to 1965 AD r_{1965}^{YiSi}, making it farther away from the pole.

To better encode the temporal information later, we further convert the polar coordinate representation to the Cartesian coordinate system, as in Eq. (3).

$$x_i = r_{AD}^{cycle} \cos(\theta_{cycle}), y_i = r_{AD}^{cycle} \sin(\theta_{cycle}) \tag{3}$$

Where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}^d$. x_i and y_i are the final representations used to encode the temporal information.

3.4 Temporal encoding

We adopt the sine and cosine functions to integrate the sexagenary year temporal information for the Transformer's (Waswani et al., 2017) position encoding. To be specific, we define the temporal encoding $TE(x_i)$ and $TE(y_i)$ for the x_i and y_i respectively, formulated as below:

$$\begin{aligned} TE(x_i): & \begin{cases} TE(x_i^{2j}) = \sin(\frac{x_i}{10000^{\frac{2j}{d}}}) \\ TE(x_i^{2j+1}) = \cos(\frac{x_i}{10000^{\frac{2j}{d}}}) \end{cases} \\ TE(y_i): & \begin{cases} TE(y_i^{2j}) = \sin(\frac{y_i}{10000^{\frac{2j}{d}}}) \\ TE(y_i^{2j+1}) = \cos(\frac{y_i}{10000^{\frac{2j}{d}}}) \end{cases} \end{aligned} \tag{4}$$

The temporal encodings $TE(x_i)$ and $TE(y_i)$ are added to the original input embedding h_i, forming a new temporal enhanced input embedding $h_i^{'} \in \mathbb{R}^{l \times d}$ that includes sexagenary cycle chronological information, defined as below:

$$h_i^{'} = h_i + TE(x_i) + TE(y_i) \tag{5}$$

As shown in Exp2 and Exp3 in Figure 3(b), despite their high similarity in the semantic representation of input embeddings, there exist distinct differences due to variations in time information. With the temporal encoding of sexagenary year, it is more prone to capturing this difference.

3.5 Temporal alignment of LLMs' weight space

Using the sexagenary cycle chronology to represent temporal information encoded in the input hidden embeddings, we propose a temporal alignment objective to further post-train the LLMs, transforming the LLMs' pre-trained weight space to a temporally enhanced new one and allowing the LLMs' representation to establish a linkage between learned knowledge and the related time period.

Given a set of n hidden embeddings $H = \{h_1^{'}, h_2^{'}, ..., h_n^{'}\}$ generated from the n input sequences, each hidden embedding $h_i^{'}$ is constructed using the temporal encoding module's Eq. (5). Through post-training on time-related texts, we aim to further transform the existing weight space θ_G of the trained LLM M with general tasks into a time alignment weight space θ_T, thereby enhancing the connection between the temporal information and learned knowledge. The definition of the transformation between weight spaces is as follows:

$$L_{temporal}: M(h_i^{'}; \theta^G) \rightarrow M(h_i^t; \theta^T) \tag{6}$$

Where $h_i^t \in H^t = \{h_1^t, h_2^t, ..., h_n^t\}$ represents the temporally aligned embeddings. θ^G and θ^T are the LLMs' pre-trained weight space and the one

transformed after the temporal alignment objective function $L_{temporal}$.

To define the temporal alignment function $L_{temporal}$, we apply Elastic Weight Consolidation (EWC) theory (Kirkpatrick et al., 2017), which is proposed to find a solution to a new task in the vicinity of an older one. In our scenario, EWC protects the general capabilities of the LLMs (simplified as Task G) by constraining the parameters θ^T of time-related tasks (simplified as Task T) utilizing a quadratic penalty to stay in a region of low error for the prior general task G centered around its parameters θ^G. According to EWC's theory, $L_{temporal}$ is defined as below:

$$\mathcal{L}_{temporal} = \mathcal{L}_T(\theta^T) + \lambda\mathcal{L}[(\theta^G) \rightarrow (\theta^T)] \quad (7)$$

$\mathcal{L}_T(\theta^T)$ is the loss for the task T only. Task T necessitates enhancement in the LLMs by transforming the weight space $\theta^G \rightarrow \theta^T$, while preserving the prior general knowledge of the LLMs. λ sets how important the old task is compared to the new one.

To improve the LLMs' links between already memorized general knowledge and encoded sexagenary cycle temporal information, task T uses the similarity algorithm to reassemble the hidden weight space based on the 60 categories in a sexagenary cycle. The setup is founded on Nylund et al. (2023)'s findings that "years or months that are closer together in time yield their embeddings that are also closer together in weight space."

Specifically, there are a total of $\{1, 2.., K\}(K = 60)$ sexagenary year classes, and each h_i^t is assigned to one of these class k based on t_i^{cycle}. For the embeddings H_t^k in the class k, $H_t^k = \{h_1^t, h_2^t, ..., h_m^t\}(m < n, H_t^k \subseteq H^t)$, our goal is to minimize the distance between embeddings in the intra-class while maximizing the separation between embeddings in the inter-classes, thus the objective of task T is defined as:

$$\begin{aligned} \mathcal{L}_T &= \delta\mathcal{L}_{intra} + (1-\delta)\mathcal{L}_{inter} \\ \mathcal{L}_{intra} &= 1 - \sum_{h_i^t \in H_t^k} \sum_{h_j^t \in H_t^k} cos\_sim(h_i^t, h_j^t) \\ \mathcal{L}_{inter} &= \sum_{h_i^t \in H_t^k} \sum_{h_j^t \notin H_t^k} cos\_sim(h_i^t, h_j^t) \end{aligned} \quad (8)$$

Based on Eq. (7), the target for transformation between weight spaces is to minimize the objective below:

$$\mathcal{L}_{temporal} = \mathcal{L}_T + \frac{\lambda}{2}\mathcal{F}(\theta^T - \theta^G)^2 \quad (9)$$

Where $\mathcal{F}$ is considered as the Fisher information matrix (Fisher, 1922), for it is thus easy to calculate even for large models.

The ultimate loss L_{final} comprises the generating objective of the LLMs (predicting the next token) L_{NTP} and the temporal alignment objective $L_{temporal}$, defined as below:

$$\mathcal{L}_{final} = \mathcal{L}_{NTP} + \sigma\mathcal{L}_{temporal} \quad (10)$$

Where σ controls the influence of the sexagenary cycle.

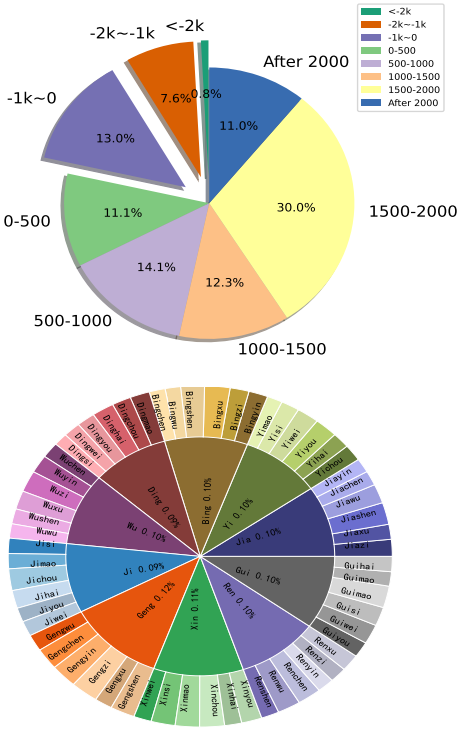


Figure 4: The distribution of the years in our constructed TempLS dataset. The above figure summarizes the distribution of Gregorian years. The figure below displays the distribution of sexagenary years, which is apparently more uniform.

4 Experiments

4.1 Datasets and downstream tasks

To evaluate the LLMs' ability to understand temporal information, we utilize two typical temporal question-answering (QA) downstream tasks: **TempLAMA** (Dhingra et al., 2022) and **TempUN** (Beniwal et al., 2024). TempLAMA is a time-sensitive QA dataset constructed based on Wikidata. TempUN is a large temporal multiple-choice QA dataset constructed by curating temporal information from the "Our World in Data" website.

Table 1: Zero-shot and few-shot (5-shot) results of LLMs measured on TempLAMA, TempUN, and TempLS. Best performance is marked as bold. **(w/ PT)**: post-train base model with the predict-next prediction objective. **(w/ Ticktack)** is the temporal enhanced model with our proposal.

Tasks		TempLS		TempLAMA				TempUN	
		Zero-shot	Few-shot	Zero-shot		Few-shot		Zero-shot	Few-shot
Model		Acc.	Acc.	ROUGE	F1	ROUGE	F1	Acc.	Acc.
Qwen2.5-3B	Base	62.37	67.81	17.13	7.45	17.14	7.45	59.25	58.88
	w/ PT	67.30	**68.29**	53.76	33.36	53.73	33.43	44.82	28.05
	w/ Ticktack	**67.63**	67.62	**54.65**	**36.43**	**54.66**	**36.44**	**64.96**	**59.92**
Qwen2.5-7B	Base	73.66	72.89	14.12	6.66	14.12	6.66	57.20	**75.60**
	w/ PT	78.12	78.17	50.97	27.19	50.97	27.19	67.79	58.22
	w/ Ticktack	**82.82**	**83.29**	**54.82**	**27.41**	**54.82**	**28.85**	**74.29**	74.37
LLaMA2-7B	Base	21.54	48.52	10.14	4.50	10.14	4.50	15.65	13.31
	w/ PT	37.61	51.22	37.09	18.44	37.09	18.44	16.33	11.84
	w/ Ticktack	**58.45**	**59.18**	**45.42**	**23.90**	**45.42**	**23.90**	**25.18**	**25.14**
TimeLLaMA-7B	Base	24.46	43.59	0.00	0.00	0.00	0.00	14.24	22.44
	w/ PT	43.54	47.61	34.95	22.61	35.82	23.06	18.84	18.89
LLaMA2-13B	Base	32.63	33.97	13.99	6.22	13.99	6.22	20.88	**26.07**
	w/ PT	24.08	43.21	55.55	27.67	55.55	27.67	12.73	22.40
	w/ Ticktack	**65.28**	**70.18**	**62.63**	**32.96**	**62.63**	**32.96**	**25.36**	25.36
TimeLLaMA-13B	Base	52.55	53.71	0.00	0.00	0.00	0.00	24.78	24.37
	w/ PT	53.00	54.24	43.77	24.40	43.77	24.40	23.85	14.53

To evaluate Ticktack's performance on the long-span time challenge, we create **TempLS**, as existing benchmarks predominantly focus on the internet age, to the best of our knowledge. **TempLS** is a long-span Chinese time-related multiple-choice QA dataset, including 137,090 question-answer pairs extracted by following the below steps. Firstly, time-related texts covering a time span from 75,000 BCE to 2025 AD are filtered from the Baidu Baike. Then the filtered texts are summarized and converted into a QA format using Qwen57B[4]. The specific distribution with the years within the dataset is depicted in Figure 4. It is apparent that with sexagenary year representation, the QA pairs have a more uniform distribution.

Appendix A.1 provides detailed information and samples of the aforementioned datasets.

4.2 Experimental setups

Baselines models. To demonstrate the generality of our method, we select several representative open-source LLMs as base models, including Qwen2.5-3B[4], Qwen2.5-7B[4](Yang et al., 2024), LLaMA2-7B[5], LLaMA2-13B[5](Touvron et al., 2023b).

We post-train the above models with our proposed Ticktack for the temporal alignments of base LLMs, denoted as **(w/ Ticktack)**, and then evaluate their performance on the three temporal QA tasks mentioned above. For TempLS, we split the post-training dataset into 132,830 training samples and 4,260 testing samples; TempLAMA and TempUN follow their existing splits.

To comparison, we post-train all base LLMs with the typical next token prediction objective, referred to as **(w/ PT)**, using the exact same post-training dataset as the Ticktack post-training. We also compare with the open-source LLM series **TimeLlama-7b** and **TimeLlama-13b**(Yuan et al., 2024), which are optimized for temporal reasoning utilizing LlaMA2.

Implementation details. We employ the parameter expansion technique, known as LoRA (Hu et al., 2021), for the post-training strategy of Ticktack. We are freezing the pre-trained parameters of the LLMs while incorporating trainable rank-decomposition matrices into each layer. α_{AD} and β_{AD} in Eq. (3) are set to [0.5-1.0] and [0.5-1.0] for changes of them within this range similarly affect the value of the Cartesian coordinate according to our experiments. σ in Eq. (8) is set to 0.5. δ in

[4] https://github.com/QwenLM/Qwen2.5

[5] https://ai.meta.com/resources/models-and-libraries/llama/

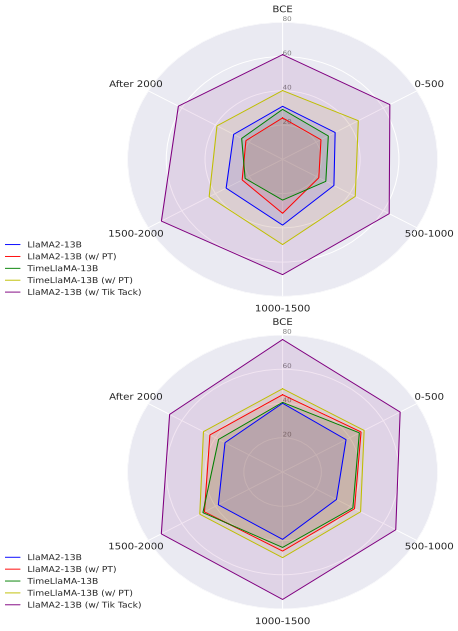


Figure 5: Accuracy of Zero-Shot and Few-shot evaluations on the TempLS for the time-span from years BCE to after 2000.

Eq. (10) is affected by the batch size and set to 1 in current experiments. The parameter study can be found in the Appendix A.5. The hyperparameters employed for all base models are as follows: a batch size of 8, gradient accumulation steps of 2, 10 epochs, and a learning rate of 10^{-4}. During the validation of downstream tasks, we utilize zero-shot and 5-shot settings to evaluate the best performance of the models. Our setup consists of a four-core CPU and eight NVIDIA Tesla A100 GPUs.

4.3 Results and analysis

Performance on downstream tasks. Table 1 presents the zero-shot and few-shot experimental results for the three temporal downstream tasks. Notably, each model trained using our proposed Ticktack significantly outperforms its baseline and post-trained counterparts across various performance metrics. In comparison to post-training alone (w/ PT), the model trained with Ticktack (w/ Ticktack) exhibits remarkable enhancements across nearly all measures in the three downstream tasks. Ticktack demonstrates an average accuracy increase of 34.43% on TempLS in comparison with the base LLaMA2-13B. In contrast to the comparative temporal enhanced baselines TimeLLaMA-7B and TimeLLaMA-13B, our method for both 7B and 13B scales achieves better results on most evaluation metrics. Furthermore, due to its adaptability with the format of the TempLAMA task (TempLAMA is the only task whose answer is not choice, detailed in Table 3 of AppendixA), the evaluation results of TimeLLaMa models on this task are 0. After post-training, the performance of TimeLLaMa models could improve on TempLAMA. Particularly, Ticktack demonstrates a more significant enhancement on long-span datasets TempLS compared to most of the baseline models.

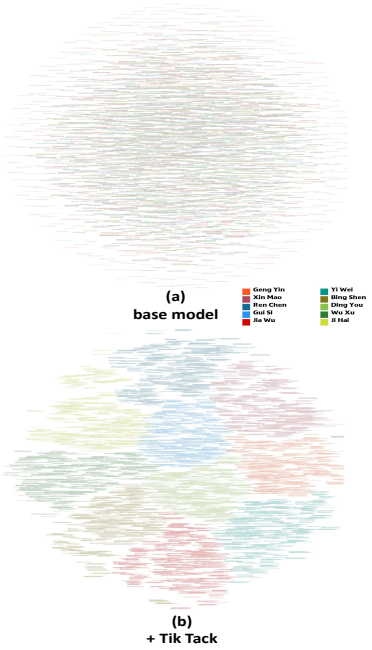
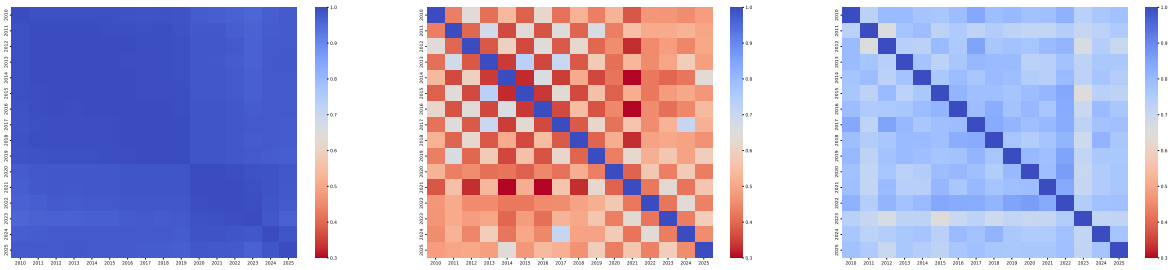


Figure 6: The distribution of Qwen-3B's outputs of 10,000 sentence vectors. (a) The pre-trained embeddings are dispersed throughout the vector space, which is hard to distinguish. (b) After post-training by Ticktack, the temporal enhanced embeddings exhibit clustering characteristics according to years.

Performance on low resource years. To analyze temporal reasoning ability in low-resource years, we separated the TempLS dataset into 500-year intervals, in addition to BCE and after 2000 periods, as shown in Figure 4. We hypothesize that the years of the BCE period have a comparatively low frequency of occurrence due to the long duration of seventy thousand five hundred years. As illustrated in Figure 5, Ticktack achieves more significant enhancement in the low-resource years of BCE, intuitively displayed through the yellow line.

Visualization of sexagenary year representations. We use T-SNE (Van der Maaten and Hinton, 2008) to visualize the multi-dimensional embeddings of Qwen-3B's outputs before and after temporal alignment with Ticktack, as illustrated



(a) Base model (b) Temporal encoding (c) Ticktack temporal aligned

Figure 7: Similarity between different years' representations, where years are chosen from 2010 to 2025. As the color of the cell approaches orange, it indicates a lower similarity between these two years' representations. (a) The embeddings of Gregorian years are generated by the base Qwen 3B. (b) Sexagenary year time encoding $TE_(x_i)$ and $TE_(y_i)$ proposed in Eq. (4), without post training. (c) Using the Ticktack post-trained Qwen-3b w/Ticktack to generate the representations of Gregorian years.

in Figure 6. We sample 10,000 sentence vectors across ten-year periods from TempLAMA ("Geng Yin to Ji Hai"). It is distinctly evident that Ticktack-enhanced embeddings in Figure 6(b) exhibit clustering characteristics according to years. The pre-trained embeddings in Figure 6(a), on the other hand, are dispersed throughout the vector space, which may make the model more susceptible to temporal reasoning errors.

More experiments and analyses can be found in the Appendix A.3 and A.4.

Table 2: Zero-shot and few-shot results of LLMs measured on TempLS, by adding temporal encoding module.

Tasks		TempLS	
		Zero-shot	Few-shot
Model		Acc.	Acc.
Qwen2.5-3B	Base	62.37	67.81
	w/ encoding	66.55(+4.18)	66.13(-1.68)
Qwen2.5-7B	Base	73.66	72.89
	w/ encoding	73.89(+0.23)	79.94(+7.05)
LLaMA2-7B	Base	21.54	48.52
	w/ encoding	30.58(+9.04)	50.98(+2.46)
LLaMA2-13B	Base	32.63	20.88
	w/ encoding	30.68(-1.95)	35.31(+14.43)

4.4 Ablation study

Similarity between different years representations. Figure 7 depicts the distinguishability of the representation of years (2010-2025), generated by LLMs. Figure 7(a) shows the representation similarity of Gregorian years from the Qwen 3B base, which has similar characteristics, making it difficult for the model to distinguish. Alternatively, our proposed sexagenary year expression with polar coordinate time encoding introduces enriched temporal information into the LLM, resulting in a clear distinction as depicted in Figure 7 (b). After post-training with Ticktack, the temporally aligned LLM improves its sensitivity to temporal information, as seen in Figure 7 (c). In comparison to the original base model, time embeddings become more discriminable, making LLMs more likely to recognize, contributing to increased performance on time-sensitive tasks.

The impact of temporal encoding module. To investigate the effect of the sexagenary year expression and encoding, we post train the LLMs by adopting temporal encoding of the polar coordinate represented sexagenary year (w/ encoding) with predict next token target that does not use temporal alignment. As shown in Table 2, only the sexagenary year's temporal encoding facilitates the enhancement of time-sensitive reasoning capabilities.

5 Conclusion

In this study, we focus on addressing the temporal misalignment issues that often affect LLMs when dealing with long-span temporal information. We first introduce the sexagenary-cycle time expression leveraging the polar coordinate to provide a more uniform and consistent temporal embedding expression. Furthermore, a temporal alignment method is proposed to enhance the LLMs' alignment of learned knowledge to the related time period. Experimental results have validated the effectiveness of our method, demonstrating its ability to enhance the performance of LLMs in handling time-related tasks with long temporal spans.

6 Limitations

While we use the sexagenary cycle time expression to align long-term temporal data, we only consider the year granularity. We will explore using the sexagenary cycle to represent many granularities of time, such as month and day, in the future. In addition, since there are few long-span time-related benchmarks, we developed the TempLS dataset, which we collected from Baidu Baike, to measure LLMs' understanding of temporal information. The relatively small number of samples may limit the generalizability and robustness of the evaluation results. It would be interesting to develop novel benchmarks derived from a range of sources to increase the comprehensiveness and reliability of temporal information understanding assessments.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Oshin Agarwal and Ani Nenkova. 2022. Temporal effects on pre-trained models for language processing tasks. *Transactions of the Association for Computational Linguistics*, 10:904–921.

Himanshu Beniwal, Dishant Patel, Kowsik Nandagopan D, Hritik Ladia, Ankit Yadav, and Mayank Singh. 2024. Remember this event that year? assessing temporal information and understanding in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16239–16348, Miami, Florida, USA. Association for Computational Linguistics.

Hsuvas Borkakoty and Luis Espinosa-Anke. 2024. Chew: A dataset of changing events in wikipedia. *arXiv preprint arXiv:2406.19116*.

Angel X. Chang and Christopher Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3735–3740, Istanbul, Turkey. European Language Resources Association (ELRA).

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Felix Drinkall, Eghbal Rahimikia, Janet B Pierrehumbert, and Stefan Zohren. 2024. Time machine gpt. *arXiv preprint arXiv:2404.18543*.

Ronald A Fisher. 1922. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604):309–368.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew

Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang,

Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. abs/2407.21783.

Wes Gurnee and Max Tegmark. 2023. Language models represent space and time. *arXiv preprint arXiv:2310.02207*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Kokil Jaidka, Niyati Chhaya, and Lyle Ungar. 2018. Diachronic degradation of language models: Insights from social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 195–200.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2022. TemporalWiki: A lifelong benchmark for training and evaluating ever-evolving language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6237–6250, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. 2021. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. 2022. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland. Association for Computational Linguistics.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A. Smith. 2022. Time waits for no one! analysis and challenges of temporal misalignment. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5944–5958, Seattle, United States. Association for Computational Linguistics.

Katerina Margatina, Shuai Wang, Yogarshi Vyas, Neha Anna John, Yassine Benajiba, and Miguel Ballesteros. 2023. Dynamic benchmarking of masked language models on temporal concept drift with multiple views. *arXiv preprint arXiv:2302.12297*.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L Griffiths. 2023. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Kai Nylund, Suchin Gururangan, and Noah A Smith. 2023. Time is encoded in the weights of finetuned language models. *arXiv preprint arXiv:2312.13401*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 840–854, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Yuqing Wang and Yun Zhao. 2023. Tram: Benchmarking temporal reasoning for large language models. *arXiv preprint arXiv:2310.00835*.

A Waswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A Gomez, L Kaiser, and I Polosukhin. 2017. Attention is all you need. In *NIPS*.

Yuting Wei, Meiling Li, Yangfu Zhu, Yuanxing Xu, Yuqing Li, and Bin Wu. 2025. A diachronic language model for long-time span classical chinese. *Information Processing & Management*, 62(1):103925.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Sen Yang, Xin Li, Lidong Bing, and Wai Lam. 2023. Once upon a *time* in *graph*: Relative-time pretraining for complex temporal reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11879–11895, Singapore. Association for Computational Linguistics.

Chenhan Yuan, Qianqian Xie, Jimin Huang, and Sophia Ananiadou. 2024. Back to the future: Towards explainable temporal reasoning with large language models. In *Proceedings of the ACM on Web Conference 2024*, pages 1963–1974.

Michael Zhang and Eunsol Choi. 2023. Mitigating temporal misalignment by discarding outdated facts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14213–14226, Singapore. Association for Computational Linguistics.

Bowen Zhao, Zander Brumbaugh, Yizhong Wang, Hannaneh Hajishirzi, and Noah Smith. 2024. Set the clock: Temporal alignment of pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15015–15040, Bangkok, Thailand. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

A Appendix

A.1 Datasets

TempLAMA (Dhingra et al., 2022): a time-sensitive question-answering dataset constructed based on the Wikidata temporal KB, is proposed to evaluate the model's performance for time-dependent questions from 2010 to 2020. The generated answers of LLMs are evaluated by token-level mirco-F1 and ROUGE-1 scores.

TempUN (Beniwal et al., 2024): a large temporal QA dataset constructed by curating temporal information from "Our World in Data" website. TempUN is used to explore the model's ability to grasp factual knowledge, containing data for global issues like poverty, disease, hunger, climate change, war, existential risks, and inequality from 10,000 BCE to 2100 AD. The format of this task is a multiple choice question answering, with accuracy serving as the evaluation metric.

Samples of the three datasets are illustrated in Table 3.

A.2 Visualization of sexagenary cycle time expression

We also utilize T-SNE to project the multi-dimensional embeddings of LLaMA2-13B and Deepseek-7B's outputs before and after temporal adaptation with Ticktack into a two-dimensional space for visualization.

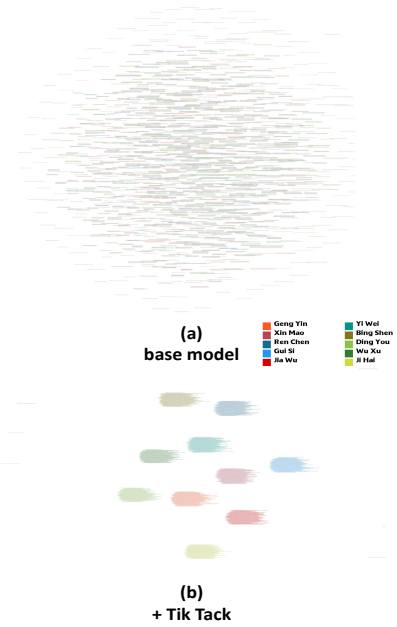


Figure 8: The distribution of LLaMA2-13B's outputs of 10,000 sentence vectors, before and after post training by Ticktack.

As shown in Figure 8 and 9, the expressions of all models also exhibit characteristics of aggregation based on the sexagenary cycle after post-training used Ticktack. However, compared to Qwen-3B displayed in Figure 6, the intra-class

Table 3: Samples from TempLS, TempLAMA and TempUN.

Dataset	Sample	
TempLS	Question:	In 3100 BC, ____ established the First Dynasty of Ancient Egypt. A: Menes, B: Ramses, C: Tutankhamun, D: Cleopatra
	Answer:	A
TempLAMA	Question:	In 2017, Alexander Hamilton is owned by _X_.
	Answer:	Crystal Bridges Museum of American Art
TempUN	Question:	Which option is correct for the question: In 2022, Private Civil Liberties Index in Iran was: Options: A: 0.49, B: 0.38, C: 0.63, D: 0.34
	Answer:	D

distance in LLaMA2-13B is more concentrated, while the distance between inter-classes is more dispersed. This may be due to the different training data adopted by different LLMs.

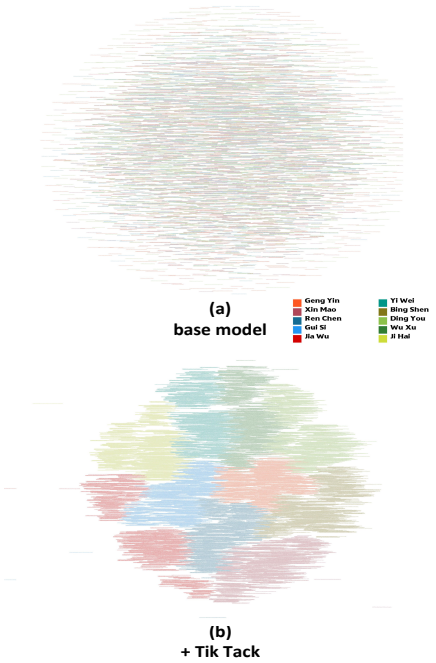


Figure 9: The distribution of Deepseek-7B's outputs of 10,000 sentence vectors, before and after post training by Ticktack.

A.3 Multilingual Experiment

In addition to Chinese and English, we also conducted basic mulitiligual (Japanese and French) evaluations comparing three model variants - the base, w/PT, and w/TickTack models - across different language pairs maintaining consistent experimental settings. The Japanese test dataset was obtained by translating TempUN, and the French test dataset was from https://huggingface.co/datasets/ab-iitd/mTEMPREASON/tree/main. As shown in table 4 and table 5, w/ PT and w/ TikTack achieved reasonably competitive results on Japanese while showing significant improvements on French, proving TickTack's effectiveness in multilingual tasks.

Table 4: Zero-shot and few-shot results of Qwen2.5-7B measured on Japanese TempUN Dataset.

Model		Zero-shot	Few-shot
Qwen2.5-7B	base	51.79	73.37
	w/PT	66.95	67.56
	w/Ticktack	64.84	66.39

Table 5: Zero-shot and few-shot results of LLaMA measured on French TEMPREASON Dataset.

Model		Zero-shot	Few-shot
LLaMA2-7B	base	23.97	51.73
	w/PT	27.27	43.29
	w/Ticktack	44.43	48.76
LLaMA3-8B	base	51.83	57.84
	w/PT	51.19	55.24
	w/Ticktack	69.16	69.16

A.4 The experimental results based on LLaMA3

To validate the adaptability of our method on LLM, we compared the performance on LLaMA3-8B (Grattafiori et al., 2024), as shown in Table 6, further demonstrating the effectiveness of our method in enhancing time intensive knowledge inference.

Table 6: Zero-shot and few-shot (5-shot) results of LLaMA3-8B measured on TempLAMA, TempUN, and TempLS. Best performance is marked as bold.

Tasks		TempLS		TempLAMA				TempUN	
		Zero-shot	Few-shot	Zero-shot		Few-shot		Zero-shot	Few-shot
Model		Acc.	Acc.	ROUGE	F1	ROUGE	F1	Acc.	Acc.
LLaMA3-8B	Base	57.96	62.42	15.97	6.43	15.97	6.43	57.96	62.42
	w/ PT	65.56	64.34	**60.83**	26.42	**60.83**	26.42	58.73	62.93
	w/ Ticktack	**68.32**	**68.71**	59.57	**37.84**	59.57	**37.84**	**65.29**	**65.24**

Table 7: Parameter study of α_{AD} and β_{AD}.

Hyper-paramters	Value	Euclidean distance
$(\alpha_{AD}, \beta_{AD})$	(0.5,0.5)	$\text{dist}_x(2025- > 2024) = 2.3151, dist_y(2025- > 2024) = 2.0244$ $\text{dist}_x(2025- > 1965) = 1.4738, dist_y(2025- > 1965) = 0.5874$
	(0.8,0.6)	$\text{dist}_x(2025- > 2024) = 1.6713, dist_y(2025- > 2024) = 1.9204$ $\text{dist}_x(2025- > 1965) = 0.8154, dist_y(2025- > 1965) = 0.7003$
	(1,1)	$\text{dist}_x(2025- > 2024) = 2.4221, dist_y(2025- > 2024) = 0.6197$ $\text{dist}_x(2025- > 1965) = 2.1998, dist_y(2025- > 1965) = 1.1240$

Table 8: Parameter study of λ, σ and δ.

λ	σ	δ	Final Training Loss
1	1	0.5	1.1177
1	1	0.3	1.121
0.5	1	0.3	1.1146
0.5	1	0.3	1.1318

A.5 Experimental parameter study

Table 7 and table 8 show the parameter study on the effect of each parameter utilized in our experiments.