Date Fragments: A Hidden Bottleneck of Tokenization for Temporal Reasoning

Anonymous ACL submission

Abstract

Modern BPE tokenizers often split calendar dates into meaningless fragments, e.g., "20250312" → "202", "503", "12", inflating token counts and obscuring the inherent structure needed for robust temporal reasoning. In this work, we (1) introduce a simple yet interpretable metric, termed date fragmentation ratio, that measures how faithfully a tokenizer preserves multi-digit date components; (2) release DATEAUGBENCH, a suite of 6500 examples spanning three temporal reasoning tasks: context-based date resolution, format-invariance puzzles, and date arithmetic across historical, contemporary, and future regimes; and (3) through layer-wise probing and causal attention-hop analyses, uncover an emergent date-abstraction mechanism whereby large language models sequentially assemble the fragments of month, day, and year components into a unified "date" concept. Our experiments show that excessive fragmentation correlates with accuracy drops of up to 10 points on uncommon dates like historical and futuristic dates. Further, we find that the larger the model, the more quickly the emergent date abstraction that heals date fragments is accomplished. Lastly, we observe a reasoning path that LLMs follow to interpret dates, relying on subword fragments that statistically represent year, month and day, and stitch these fragments in a flexible order that is subject to date formats.

1 Introduction

Understanding and manipulating dates is a deceptively complex challenge for modern large language models (LLMs). Unlike ordinary words, dates combine numeric and lexical elements in rigidly defined patterns—ranging from compact eight-digit strings such as `20250314` to more verbose forms like "March 14, 2025" or locale-specific variants such as "14/03/2025." Yet despite their structured nature, these date expressions often fall prey to subword tokenizers that fragment them into semantically meaningless pieces. A tokenizer that

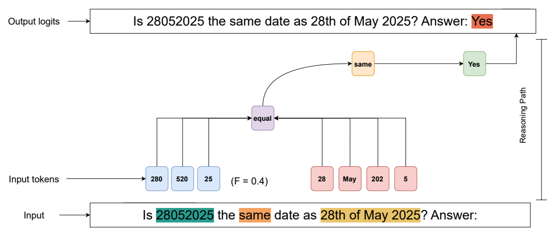


Figure 1: Internal processing of dates for temporal reasoning. Here F=0.4 shows the date fragmentation ratio.

splits "2025-03-14" into "20", "25", "-0", "3", "-1", "4" not only inflates the token count but also severs the natural boundaries of year, month, and day. This fragmentation obscures temporal cues and introduces a hidden bottleneck: even state-of-the-art LLMs struggle to resolve, compare, or compute dates accurately when their internal representations have been so badly fragmented. This issue is critical for real-world applications:

Mis-tokenized dates can undermine scheduling and planning workflows, leading to erroneous calendar invites or appointments (Vasileiou and Yeoh, 2024). They can skew forecasting models in domains ranging from time-series analysis (Tan et al., 2024; Chang et al., 2023) to temporal knowledge graph reasoning (Wang et al., 2024). In digital humanities and historical scholarship, incorrect splitting of date expressions may corrupt timelines and misguide interpretative analyses (Zeng, 2024). As LLMs are increasingly deployed in cross-temporal applications, such as climate projection(Wang and Karimi, 2024), economic forecasting (Carriero et al., 2024; Bhatia et al., 2024), and automated curriculum scheduling (Vasileiou and Yeoh, 2024), the brittleness introduced by subword fragmentation poses a risk of propagating temporal biases and inaccuracies into downstream scientific discoveries and decision-making systems (Tan et al., 2024).

In this work, we provide a pioneer outlook on the impact of date tokenization on downstream temporal reasoning. Figure 1 illustrates how dates are processed internally for temporal reasoning. Our contributions are summarized as follows:

(i) We introduce DATEAUGBENCH, a benchmark dataset comprising 6,500 examples with 21 date formats. It is leveraged to evaluate a diverse array of LLMs from 8 model families in three temporal reasoning tasks.
(ii) We present date fragmentation ratio, a metric that measures how fragmented the tokenization outcome is compared to the actual year, month, and day components. We find that the fragmentation ratio generally correlates with temporal reasoning performance, namely that the more fragmented the tokenization, the worse the reasoning performance.
(iii) We analyse internal representations by tracing how LLMs "heal" fragmented date embeddings in their layer stack—an emergent ability that we term *date abstraction*. We find that larger models quickly stitch fragmented date inputs into a unified "date" concept for temporal reasoning at early layers.
(iv) We leverage causal analysis to interpret how LLMs understand dates. Our results show that LLMs follow a reasoning path that is typically not aligned with human interpretation (year $\rightarrow$ month $\rightarrow$ day), but relies on subword fragments that statistically represent year, month, and day, and stitch them in a flexible order.

Our work fills the gap between tokenisation research (Goldman et al., 2024; Schmidt et al., 2024) and temporal reasoning (Su et al., 2024; Fatemi et al., 2024), and motivates the design of date-aware vocabularies and adaptive tokenizers that preserve temporal coherence without sacrificing numeric flexibility for future models.

2 Related Works

Tokenisation as an information bottleneck. Recent scholarship interrogates four complementary facets of sub-word segmentation: (i) *tokenisation fidelity*, i.e. how closely a tokenizer preserves semantic units: Large empirical studies show that higher compression fidelity predicts better downstream accuracy in symbol-heavy domains such as code, maths and dates (Goldman et al., 2024; Schmidt et al., 2024); (ii) *numeric segmentation strategies* that decide between digit-level or multi-digit units: Previous work demonstrates that the choice of radix—single digits versus 1–3-digit chunks—induces stereotyped arithmetic errors and can even alter the complexity class of the computations LLMs can realise (Singh and Strouse, 2024; Zhou et al., 2024); (iii) *probabilistic or learnable tokenisers* whose segmentations are optimised jointly with the model: Theory frames tokenisation as a stochastic map whose invertibility controls whether maximum-likelihood estimators over tokens are consistent with the underlying word distribution (Gastaldi et al., 2024; Rajaraman et al., 2024) and (iv) *pre-/post-tokenisation adaptations* that retrofit a model with a new vocabulary: Zheng et al. (2024) introduce an *adaptive tokenizer* that co-evolves with the language model, while Liu et al. (2025) push beyond the "sub-word" dogma with *SuperBPE*, a curriculum that first learns subwords and then merges them into cross-whitespace "superwords", cutting average sequence length by 27 %. Complementary studies expose and correct systematic biases introduced by segmentation (Phan et al., 2024) and propose *trans-tokenization* to transfer vocabularies across languages without re-training the model from scratch (Remy et al., 2024). Our work builds on these insights but zooms in on calendar dates—a hybrid of digits and lexical delimiters whose multi-digit fields are routinely shredded by standard BPE, obscuring cross-field regularities crucial for temporal reasoning.

Temporal reasoning in large language models. Despite rapid progress on chain-of-thought and process-supervised reasoning, temporal cognition remains a conspicuous weakness of current LLMs. Benchmarks such as TIMEBENCH (Chu et al., 2024), TEMPREASON (Tan et al., 2023), TEST-OF-TIME (Fatemi et al., 2024), MENATQA (Wei et al., 2023) and TIMEQA (Chen et al., 2021) reveal large gaps between model and human performance across ordering, arithmetic and co-temporal inference. Recent modelling efforts attack the problem from multiple angles: temporal-graph abstractions (Xiong et al., 2024), instruction-tuned specialists such as TIMO (Su et al., 2024), pseudo-instruction augmentation for multi-hop QA (Tan et al., 2023), and alignment techniques that reground pretrained models to specific calendar years (Zhao et al., 2024). Yet these approaches assume a faithful internal representation of the input dates themselves. By introducing the notion of *date fragmentation* and demonstrating that heavier fragmentation predicts up to ten-point accuracy drops on DATEAUGBENCH, we uncover a failure mode that is *orthogonal* to reasoning algorithms or supervision: errors arise before the first transformer layer, at the level of subword segmentation. Addressing this front-end bottleneck complements, rather than competes with, existing efforts to improve temporal reasoning in LLMs.

3 DateAugBench

We introduce DATEAUGBENCH, benchmark designed to isolate the impact of date tokenisation on temporal reasoning in LLMs. DATEAUGBENCH comprises 6,500 augmented examples drawn from two established sources, TIMEQA (Chen et al., 2021) and TIMEBENCH (Chu et al., 2024), distributed across three tasks splits (see Table 1). Across all the splits, our chosen date formats cover a spectrum of common regional conventions (numeric with slashes, dashes, or dots; concatenated strings; two-digit versus four-digit years) and deliberately introduce fragmentation for atypical historical (e.g. “1799”) and future (e.g. “2121”) dates. This design enables controlled measurement of how tokenization compression ratios and subsequent embedding recovery influence temporal reasoning performance.

Context-based task. In the *Context-based* split, we sample 500 question–context pairs from TIMEQA, each requiring resolution of a date mentioned in the passage (e.g. Which team did Omid Namazi play for in 06/10/1990?). Every date expression is systematically rendered in six canonical serialisations—including variants such as `MM/DD/YYYY`, `DD-MM-YYYY`, `YYYY.MM.DD` and concatenations without delimiters—yielding 3,000 examples that jointly probe tokenisation fragmentation and contextual grounding.

Simple Format Switching task. The *Simple Format Switching* set comprises 150 unique date pairs drawn from TIMEBENCH, posed as binary same-day recognition questions (e.g. “Are 20251403 and 14th March 2025 referring to the same date?”). Each pair is presented in ten different representations, spanning slash-, dash-, and dot-delimited formats, both zero-padded and minimally notated, to stress-test format invariance under maximal tokenisation drift. This produces 1,500 targeted examples of pure format robustness. We also have examples where the dates are not equivalent, complicating the task.

Date Arithmetic task. The *Date Arithmetic* split uses 400 arithmetic instances from TIMEBENCH (e.g. What date is 10,000 days before 5/4/2025?). With the base date serialised in five distinct ways—from month-day-year and year-month-day with various delimiters to compact eight-digit forms. This results in 2,000 examples that examine the model’s ability to perform addition and subtraction of days, weeks, and months under various token fragmentation.

4 Experiment Design

4.1 Date Tokenization

Tokenizers. For tokenization analysis, we compare a deterministic, rule-based *baseline tokenizer* against model-specific tokenizers. The baseline splits each date into its semantic components—year, month, day or Julian day—while preserving original delimiters. For neural models, we invoke either the OpenAI tiktoken encodings (for `gpt-4`, `gpt-3.5-turbo`, `gpt-4o`, `text-davinci-003`) or Hugging Face tokenizers for open-source checkpoints. Every date string is processed to record the resulting sub-tokens, token count, and reconstructed substrings.

Distance metric. To capture divergence from the ideal, we define a distance metric θ between a model’s token distribution and the baseline’s:

$$\theta(\mathbf{t}, \mathbf{b}) = 1 - \frac{\mathbf{t} \cdot \mathbf{b}}{|\mathbf{t}|, |\mathbf{b}|}, \tag{1}$$

where $\mathbf{t}$ and $\mathbf{b}$ are vectors of sub-token counts for the model and baseline, respectively. A larger θ indicates greater sub-token divergence.

Date fragmentation ratio. Building on θ, we introduce the *date fragmentation ratio* F, which quantifies how fragmented a tokenizer’s output is relative to the baseline. We initialise $F = 0.0$ for a perfectly aligned segmentation and apply downward adjustments according to observed discrepancies: a 0.10 penalty if the actual year/month/day components are fragmented (i.e., $\mathbf{1}_{\text{split}} = 1$) , a 0.10 penalty if original delimiters are lost (i.e., $\mathbf{1}_{\text{delimiter}} = 1$), a 0.05 penalty multiplied by the token count difference $\left(N - N_b\right)$ between a tokenizer and the baseline, and a $0.30 \times \theta$ penalty for distributional divergence. The resulting $F \in [0, 1]$ provides an interpretable score: values close to 0 denote minimal fragmentation, and values near 1 indicate severe fragmentation.

$$\begin{aligned} F = 0.10 * \mathbf{1}_{\text{split}} + 0.10 * \mathbf{1}_{\text{delimiter}} \\ + 0.05 * \left(N - N_b\right) + 0.30 * \theta \end{aligned} \tag{2}$$

Dataset and Task	# Formats	# Raw	Size	Evaluation	
				Example	**GT**
Context based	6	500	3000	Which team did Omid Namazi play for in 06/10/1990?	Maryland Bays
Date Format Switching	10	150	1500	Are 20251403 and March 14th 2025 referring to the same date?	Yes
Date Arithmetic	5	400	2000	What date is 10,000 days before 5/4/2025?	18 November 1997; 17 December 1997
Total	21	1500	6500		

Table 1: Overview and examples of task splits in DATEAUGBENCH.

This date fragmentation ratio is pivotal because tokenisation inconsistencies directly impair a model's ability to represent and reason over temporal inputs. When date strings are split non-intuitively, models face inflated token sequences and fragmented semantic cues, potentially leading to errors in tasks such as chronological comparison, date arithmetic, and context-based resolution. By quantifying fragmentation explicitly through F, we reveal hidden limitations in existing tokenizers, inform selections of robust architectures for time-sensitive applications.

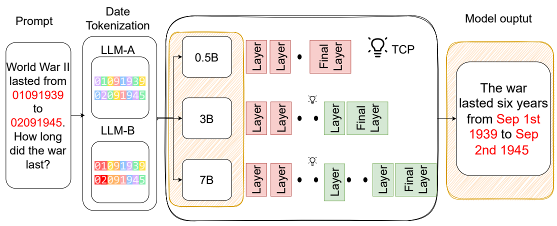


Figure 2: Illustration of how LLMs with various model sizes process dates. TCP means Tokenization Compensation Point, defined as the earliest layer at which LLMs achieve above-chance accuracy (see more details in Section 6).

4.2 Temporal Reasoning Evaluation

Models. We evaluate a spectrum of model ranging from 0.5 B to 14 B parameters: five open-source Qwen 2.5 models (0.5 B, 1.5 B, 3 B, 7 B, 14 B) (Yang et al., 2024), two Llama 3 models (3 B, 8 B) (Touvron et al., 2023b), and two OLMo (Groeneveld et al., 2024) models (1 B, 7 B). For comparison with state-of-the-art closed models, we also query the proprietary `GPT-4o` and `GPT-4o-mini` endpoints via the OpenAI API (OpenAI et al., 2024).

LLM-as-a-judge. To measure how date tokenization affects downstream reasoning, we employ an LLM-as-judge framework using GPT-4o. For each test instance in DATEFRAGBENCH, we construct a JSONL record that includes the question text, the model's predicted answer, and a set of acceptable gold targets to capture all semantically equivalent date variants (e.g., both "03/04/2025" and "April 3, 2025" can appear in the gold label set). This record is submitted to GPT-4o via the OpenAI API with a system prompt instructing it to classify the prediction as `CORRECT`, `INCORRECT`, or `NOT ATTEMPTED`. A prediction is deemed `CORRECT` if it fully contains any one of the gold target variants without contradiction; `INCORRECT` if it contains factual errors relative to all gold variants; and `NOT ATTEMPTED` if it omits the required information. We validate GPT-4o's reliability by randomly sampling 50 judged instances across all splits and obtaining independent annotations from four human reviewers. GPT-4o's classifications agree with the human consensus on 97% of cases, yielding a Cohen's κ of 0.89, which affirms the reliability of our automated evaluation.

4.3 Internal Representations

Layerwise probing. We use four Qwen2.5 (Yang et al., 2024) model checkpoints (0.5B, 1.5B, 3B, and 7B parameters) to trace how temporal information is processed internally across different layers. During inference, each question is prefixed with a fixed system prompt and a chain-of-thought cue, then passed through the model in evaluation mode. At each layer i, we extract the hidden-state vector corresponding to the final token position, yielding an embedding $h_i \in \mathbb{R}^d$ for that layer. Repeating over all examples produces a collection of layerwise representations for positive and negative cases. We then quantify the emergence of temporal reasoning by training lightweight linear probes on these embeddings. For layer i, the probe is trained to distinguish "same-date" vs "different-date" examples.

To explain when the model's date understanding is achieved, we define the *tokenization compensation point* as the layer at which the model's representation correctly represents the date in the given prompt. We experiment with this idea across various model sizes, aiming to test our hypothesis: larger models would recover calendar-level semantics from fragmented tokens at earlier stages, i.e., tokenization compensation is accomplished at early layers, as illustrated in Figure 2.

Causal attention-hop analysis. To reveal the mechanisms by which LLMs parse and resolve date strings, we conduct a two-stage causal analysis (Lindsey et al., 2025) that combines activation tracing with targeted interventions. First, we instrument the model's residual stream across all layers to capture when and where temporal information emerges. Given an input prompt requiring a date resolution (e.g., "Is 12/05/2020 the same date as 12th of May 2020?"), we identify two sets of tokens: (1) *concept tokens* corresponding to year, month, and day fragments, and (2) *decision tokens* corresponding to the final "yes" or "no" output. For each layer-wise token, we project its hidden state through the output embedding, producing an activation map whose peaks locate the layer and position that best encode each fragment or verdict. Attention peaks indicate the layer and position where temporal fragments and the judgment receive the most attention from input tokens. In the second stage, we perform causal interventions at the final transformer layer to quantify each token's influence on the model's decision. For each concept token that shows a strong activation peak, we generate a *corrupted prompt* by replacing that fragment with a contrasting value (e.g., swapping "12" for "31"). We then re-evaluate the model on the corrupted prompt and measure the change in the logit difference between "yes" and "no." The magnitude of this change reflects the causal strength of the original token's contribution to the final judgment. To build a *sparse importance map*, we multiply each token's normalised peak height by the absolute size of its causal effect. This causal framework not only pinpoints *where* and *when* temporal concepts are represented, but also *how* they sequentially combine to drive the model's final decision.

5 Experiment Results

5.1 Date fragmentation

Cross-temporal performance. Table 2 reports the mean date fragmentation ratio across four temporal regimes—*Past* (pre–2000), *Near Past* (2000–2009), *Present* (2010–2025), and *Future* (post–2025)—for each evaluated model. A ratio of 0.00 signifies perfect alignment with our rule-based baseline tokenizer, whereas higher values indicate progressively greater fragmentation. The rule-based `Baseline` unsurprisingly attains the maximal ratio of 0.00 in all periods, serving as a lower bound. Among neural architectures, OLMo (Groeneveld et al., 2024) demonstrates the highest robustness, with an average fragmentation ratio of 0.15, closely followed by GPT-3 at 0.16. Both maintain strong fidelity across temporal splits, although performance dips modestly in the Future category (0.25), reflecting novel token sequences not seen during pre-training.

Model	Past	Near Past	Present	Future	Avg
Baseline	0.00	0.00	0.00	0.00	0.00
OLMo	0.15	0.14	0.07	0.25	0.15
GPT-3	0.17	0.14	0.06	0.25	0.16
Llama 3	0.29	0.28	0.27	0.30	0.29
GPT-4o	0.32	0.31	0.22	0.30	0.29
GPT-3.5	0.47	0.22	0.26	0.36	0.33
GPT-4	0.36	0.26	0.29	0.39	0.33
Qwen	0.58	0.55	0.49	0.58	0.55
Gemma	0.58	0.55	0.49	0.58	0.55
DeepSeek	0.58	0.55	0.49	0.58	0.55
LlaMa	0.63	0.63	0.63	0.63	0.63
Phi	0.63	0.63	0.63	0.63	0.63

Table 2: Date fragmentation ratio across models and data splits over time.

Models	Context Rlt	Fmt Switch	Date Arth.	Avg.
GPT-4o-mini	53.20	95.66	56.67	68.51
OLMo-2-7B	32.13	97.24	64.72	64.70
Qwen2.5 14B	47.56	94.56	51.35	64.49
Qwen2.5 7B	39.56	91.24	40.56	57.12
Qwen2.5 3B	25.45	90.10	39.45	51.67
LLama3.1 8B	26.20	90.22	34.50	50.31
Qwen2.5 1.5B	21.32	89.65	32.34	47.77
Qwen2.5 0.5B	10.23	88.95	31.32	43.50
OLMo-2-1B	9.26	90.09	25.90	41.75
LLama3.2 3B	9.51	88.45	23.66	40.54

Table 3: Average accuracies per task. Context Rlt stands for context based resolution task, Fmt Switch refers to the format switching task, and Date Arth. refers to the date arithmetic task.

Impact of subtoken granularity. A closer look, from Table 4, at sub-token granularity further explains these trends. Llama 3 (Touvron et al., 2023b) and the GPT (OpenAI et al., 2023) families typically segment each date component into three-digit sub-tokens (e.g., "202", "504", "03"), thus preserving the semantic unit of "MMDDYYYY" as compact pieces. OLMo (Groeneveld et al., 2024)

Model	Tokenized output	Frag-ratio
Baseline	`10 27 1606`	0.00
OLMo	`10 27 16 06`	0.34
Llama 3	`102 716 06`	0.40
GPT-3	`1027 16 06`	0.40
GPT-4o	`102 716 06`	0.40
Gemma	`1 0 2 7 1 6 0 6`	0.55
DeepSeek	`1 0 2 7 1 6 0 6`	0.55
Cohere	`1 0 2 7 1 6 0 6`	0.55
Qwen	`1 0 2 7 1 6 0 6`	0.55
Phi 3.5	`_ 1 0 2 7 1 6 0 6`	0.60
Llama 2	`_ 1 0 2 7 1 6 0 6`	0.60

Table 4: Tokenisation of the MMDDYYYY string "`10271606`" across models.

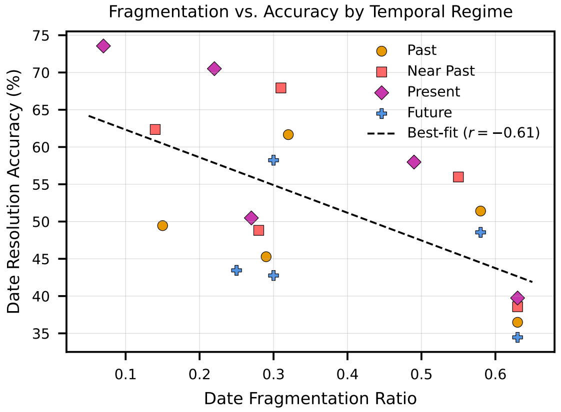


Figure 3: Date fragmentation ratio versus date resolution accuracy, stratified by temporal regime and six LLMs: OLMo, Llama 3, GPT-4o, Qwen, Gemma, Phi.

splits the date tokens into two digit tokens (e.g., "20", "25"). By contrast, Qwen (Yang et al., 2024) and Gemma (Team et al., 2024) models break dates into single-digit tokens (e.g., "2", "5"), whereas Phi (Abdin et al., 2024) and LLama (Touvron et al., 2023a) divide it into single-digit tokens with an initial token (e.g. "_", "2", "0", "2", "5"), inflating the token count. Although single-digit tokenisation can enhance models' ability to perform arbitrary numeric manipulations (by treating each digit as an independent unit), it comes at the expense of temporal abstraction: the tight coupling between day, month, and year is lost, inflating the compression penalty and increasing the θ divergence from the baseline.

5.2 DateFragBench Evaluation

Performance on temporal reasoning tasks. We compare model accuracies in three tasks: Context-based Resolution, Format Switching, and Date Arithmetic (see the results in Table 3). All models effectively solve Format Switching (e.g. 97.2% for OLMo-2-7B, 95.7% for GPT-4o-mini, 94.6% for Qwen2.5-14B, 90.2% for Llama3.1-8B). By contrast, Context Resolution and Arithmetic remain challenging: GPT-4o-mini scores 53.2% and 56.7%, Qwen2.5-14B 47.6% and 51.4%, Llama3.1-8B 26.2% and 34.5%, and OLMo-2-7B 32.1% and 64.7%, respectively. The fact that arithmetic performance consistently exceeds resolution suggests that, given a correctly tokenized date, performing addition or subtraction is somewhat easier than resolving the date within free text—which requires encyclopedic knowledge.

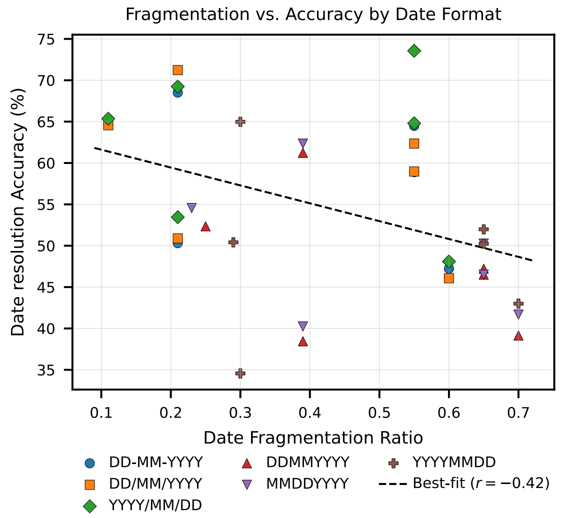


Figure 4: Date fragmentation ratio versus date resolution accuracy, stratified by six formats and six LLMs.

Correlating date fragmentation with model accuracy over time. Figure 3 plots date fragmentation ratio against resolution accuracy, with 24 data points across six models and four temporal splits. Accuracy rises as we move from Past (1600-2000) to the Near Past (2000–2009) and peaks in the Present (2010–2025), mirroring the negative correlation between fragmentation and accuracy (dashed line, Pearson correlation of -0.61). We note that the correlation is not particularly strong. This is because (i) for some models, their date fragmentation ratios remain unchanged across temporal data splits and (ii) models differ greatly by their sizes: a larger model often outperforms a substantially smaller model in resolution accuracy, even if the former has a much higher fragmentation ratio.

As seen from Table 5, GPT-4o-mini climbs from 61.7 % in the Past to 67.9 % in the Near Past, peaks at 70.5 % for Present, and falls to 58.2 % on Future dates. Qwen-2.5-14B and Llama-3.1-8B trace the same contour at lower absolute levels. OLMo-2-7B shows the steepest Near-Past jump (49.5 → 62.4 %) and achieves the highest Present accuracy (73.6 %), consistent with its finer-grained tokenisation of "20XX" patterns. These results indicate that while finer date tokenisation (i.e., lower fragmentation ratios) boosts performance up to contemporary references, today's models still generalise poorly to genuinely novel (post-2025) dates, highlighting an open challenge for robust temporal reasoning.

Correlating date fragmentation with model accuracy over formats. Figure 4 plots model accuracy against date fragmentation ratio across six date formats and six LLMs. A moderate negative trend emerges (dashed line, Pearson correlation of -0.42): formats that contain explicit separators (DD-MM-YYYY, DD/MM/YYYY, YYYY/MM/DD) are tokenised into more pieces and, in turn, resolved more accurately than compact, separator-free strings (DDMMYYYY, MMDDYYYY, YYYYMMDD). As shown in Table 6, GPT-4o-mini tops every format and receives a moderate performance drop from 71.2 % on DD/MM/YYYY to 61.2 % on DDMMYYYY, with the highest overall average (66.3 %). OLMo-2-7B and Qwen-2.5-14B both exceed 70 % on the highly fragmented YYYY/MM/DD form, but slip into the low 50s on MMDDYYYY and YYYYMMDD. Lower date fragmentation ratio models, such as Llama-3.1-8B and Phi-3.5, lag behind; their accuracy plunges below 40 %. Even so, all models score much better on separator-rich formats compared to the date formats without separators. In summary, model accuracy is correlated to how cleanly a model can tokenize the string into interpretable tokens: more visual structure (slashes or dashes) means lower fragmentation, which suggests more straightforward reasoning, and in turn, leads to better performance.

6 *When* do LLMs understand dates?

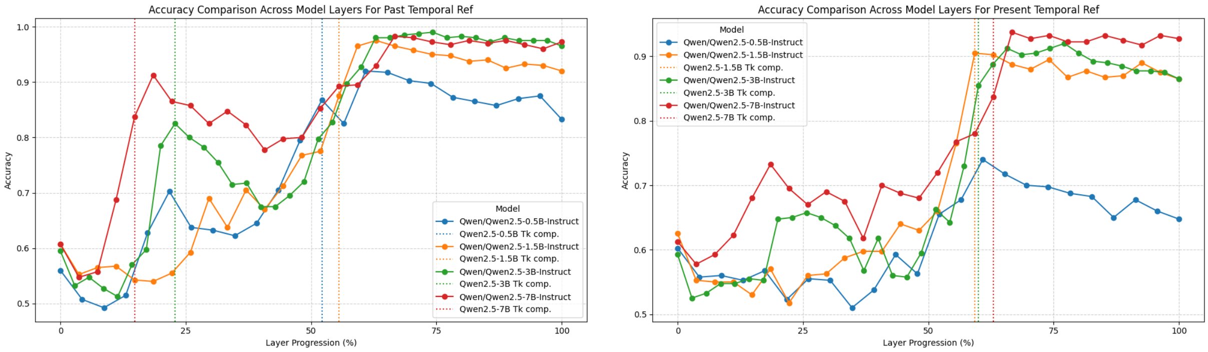
Layerwise linear probing. To pinpoint in which layer a model learns to recognize two equivalent dates, we define the *tokenization compensation point* (TCP) as the earliest layer at which a lightweight linear probe on the hidden state achieves above-chance accuracy, which is defined as 80%, on the date equivalence task. Figure 5a reports TCPs for the DATES_PAST benchmark (1600–2010): Qwen2.5-0.5B reaches TCP at layer 12 (50% depth), Qwen2.5-1.5B at layer 15 (53.6%), Qwen2.5-3B at layer 8 (22.2%), and Qwen2.5-7B at layer 4 (14.3%). The leftward shift of the 3B and 7B curves suggests how larger models recover calendar-level semantics from fragmented tokens more rapidly. Figure 5b shows the DATES_PRESENT benchmark (2010–2025), where only the 1.5B, 3B, and 7B models surpass TCP—at layers 16 (57.1%), 21 (58.3%), and 17 (60.7%), respectively—while the 0.5B model never does. The deeper TCPs here reflect extra layers needed to recombine the two-digit "20" prefix, which is fragmented unevenly by the tokenizer. In Figure 10, we evaluate DATES_FUTURE (2025–2599), where novel four-digit sequences exacerbate fragmentation. Remarkably, TCPs mirror the Past regime: layers 12, 15, 8, and 4 for the 0.5B, 1.5B, 3B, and 7B models, respectively. This parallelism indicates that model scale dictates how quickly fragmented inputs are stitched into a unified "date" concept for temporal reasoning, even when dates are novel.

Tokenization compensation point. Overall, we observe a sharp decline in TCP as model size increases: small models defer date reconstruction to middle layers, whereas the largest model does so within the first quarter of layers. Across all the three temporal benchmarks, TCP shifts steadily toward the first layers as model size grows, and its absolute position is mainly independent of date ranges being tested.

7 *How* do LLMs understand dates?

Causal path tracing. To investigate how LLMs like Llama 3 (Touvron et al., 2023b) internally understand a date string, we traced the dominant attention "hops" from the model answer token back through the input digits. Figure 6 plots model layers on the y axis against prompt tokens (e.g., Is 03122025 a valid date?) on the x axis. Green arrows mark the attention path with the highest weight that is responsible for generating the answer "yes". Activation peaks at the final layers sequentially highlight the different fragments "25", "220", "031", the abstract "date" concept, and finally the "yes" output. The model tokenizes the input into "220", "031", and "25" as tokens. The LLM understands these tokens differently from the input. This chain of token-level jumps reveals that the LLM performs a kind of discrete, step-by-step pattern aggregation, stitching together substrings of the input until a binary valid/invalid verdict emerges.

Date understanding and explainability. In contrast, human readers parse dates by immediately mapping each component to a coherent temporal schema: "03" is March, "12" is day of month,



(a) Past

(b) Present

Figure 5: Layer-wise accuracies in the two periods: Past and Present.

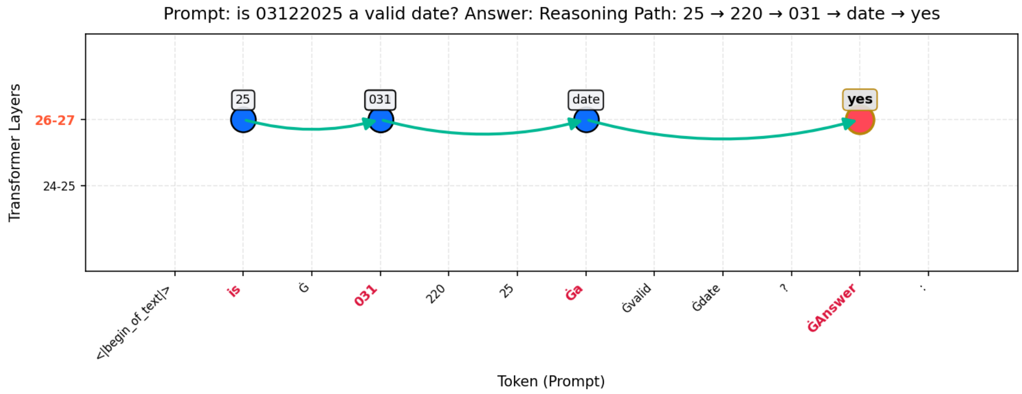


Figure 6: Causal-tracing of the “03122025 is a valid date” judgment.

“2025” is year, and then checking whether the day falls within the calendar bounds of that month. Humans bring rich world knowledge of calendars and leap-year rules to bear in parallel. However, LLMs exhibit no explicit calendar “module”; instead, they rely on learned statistical associations between digit-patterns and the training-time supervisory signal for “valid date.” The causal-tracing path in Figure 6 thus illustrates a fundamentally different mechanism of date comprehension in LLMs, based on token-level attention re-routing rather than holistic semantic interpretation. We repeated causal tracing on 100 date strings in 6 different date formats to test whether this attention trajectory is consistent across date formats. In most of cases, we observe that the model’s attention hops (i.e., reasoning paths) are not aligned with human interpretation (year $\rightarrow$ month $\rightarrow$ day), rather *sub-word fragments* that statistically represent year, month, and day in a flexible order that is subject to date formats (see examples in Figures 7-8). However, such date understanding becomes tricky when a date is greatly fragmented: given the date abstraction is learned from frequency rather than hard-coded rules, the abstraction is biased toward standard Western formats and contemporary years. As a result, a model often addresses popular dates with higher model accuracy and similar date reasoning paths. However, the reasoning path becomes obscure on rare, historical, or locale-specific strings outside the distribution of pre-training data (see Figure 9).

8 Conclusion

In this paper, we identified date tokenization as a critical yet overlooked bottleneck in temporal reasoning with LLMs. We demonstrated a correlation between date fragmentation and task performance in temporal reasoning, i.e., the more fragmented the tokenization, the worse the reasoning performance. Our layerwise and causal analyses in LLMs further revealed an emergent “date abstraction” mechanism that explains when and how LLMs understand and interpret dates. Our results showed that larger models can compensate for date fragmentation by stitching fragments into a unified “date” concept, while the stitching process appears to be accomplished via a reasoning path that connects date fragments in a flexible order, differing from human interpretation from year to month to day.

Limitations

While our work demonstrates the impact of date tokenization on LLMs for temporal reasoning, there are several limitations. First, DATEAUGBENCH focuses on a finite set of canonical date serialisations and does not capture the full diversity of natural-language expressions (e.g., “the first Monday of May 2025”) or noisy real-world inputs like OCR outputs. Second, our experiments evaluate a representative but limited pool of tokenizers and model checkpoints (up to 14B parameters); therefore, the generalizability of date fragmentation ratio and our probing and causal analyses to very large models with 15B+ parameters remains unknown. Finally, while the fragmentation ratio measures front-end segmentation fidelity, it does not account for deeper world-knowledge factors such as leap-year rules, timezone conversions, and culturally grounded calendar systems, all of which would influence temporal interpretation. Future work should extend to more diverse date expressions, broader model and tokeniser families, and equipping tokenisers with external calendar-wise knowledge to further improve robust temporal reasoning.

Ethical Considerations

DATEAUGBENCH is derived solely from the public, research-licensed TIMEQA and TIMEBENCH corpora that do not contain sensitive data; our augmentation pipeline rewrites only date strings. However, our dataset focuses on 21 Anglo-centric Gregorian formats. Therefore, our data potentially reinforce a Western default and overlook calendars or numeral systems used in many other cultures, and our date fragmentation metric may over-penalise tokenisers optimised for non-Latin digits.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone.

Gagan Bhatia, El Moatez Billah Nagoudi, Hasan Cavusoglu, and Muhammad Abdul-Mageed. 2024. Fintral: A family of gpt-4 level multimodal financial large language models. *Preprint*, arXiv:2402.10986.

Andrea Carriero, Davide Pettenuzzo, and Shubhranshu Shekhar. 2024. Macroeconomic forecasting with large language models. *arXiv preprint arXiv:2407.00890*.

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. 2023. Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *arXiv preprint arXiv:2308.08469*.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. *Preprint*, arXiv:2108.06314.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2024. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. *Preprint*, arXiv:2311.17667.

Bahare Fatemi, Mehran Kazemi, Anton Tsitsulin, Karishma Malkan, Jinyeong Yim, John Palowitch, Sungyong Seo, Jonathan Halcrow, and Bryan Perozzi. 2024. Test of time: A benchmark for evaluating llms on temporal reasoning.

Juan Luis Gastaldi, John Terilla, Luca Malagutti, Brian DuSell, Tim Vieira, and Ryan Cotterell. 2024. The foundations of tokenization: Statistical and computational concerns.

Omer Goldman, Avi Caciularu, Matan Eyal, Kris Cao, Idan Szpektor, and Reut Tsarfaty. 2024. Unpacking tokenization: Evaluating text compression and its correlation with model performance.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. Olmo: Accelerating the science of language models.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. On the biology of a large language model. *Transformer Circuits Thread*.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, and Yejin Choi. 2025. Superbpe: Space travel for language models. *arXiv preprint arXiv:2503.13423*.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2023. Gpt-4 technical report.

Buu Phan, Marton Havasi, Matthew Muckley, and Karen Ullrich. 2024. Understanding and mitigating tokenization bias in language models. *arXiv preprint arXiv:2406.16829*.

Nived Rajaraman, Jiantao Jiao, and Kannan Ramchandran. 2024. Toward a theory of tokenization in llms.

François Remy, Pieter Delobelle, Hayastan Avetisyan, Alfiya Khabibullina, Miryam de Lhoneux, and Thomas Demeester. 2024. Trans-tokenization and cross-lingual vocabulary transfers: Language adaptation of llms for low-resource nlp. *arXiv preprint arXiv:2408.04303*.

Craig W. Schmidt, Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. 2024. Tokenization is more than compression.

Aaditya K. Singh and DJ Strouse. 2024. Tokenization counts: the impact of tokenization on arithmetic in frontier llms.

Zhaochen Su, Jun Zhang, Tong Zhu, Xiaoye Qu, Juntao Li, Min Zhang, and Yu Cheng. 2024. Timo: Towards better temporal reasoning for language models.

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. 2024. Are language models actually useful for time series forecasting? In *Advances in Neural Information Processing Systems*.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards robust temporal reasoning of large language models via a multi-hop qa dataset and pseudo-instruction tuning.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Stylianos Loukas Vasileiou and William Yeoh. 2024. Trace-cs: A synergistic approach to explainable course scheduling using llms and logic. *arXiv preprint arXiv:2409.03671*.

Jiapu Wang, Kai Sun, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. 2024. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. *arXiv preprint arXiv:2405.14170*.

Yang Wang and Hassan A Karimi. 2024. Exploring large language models for climate forecasting. *arXiv preprint arXiv:2411.13724*.

Yifan Wei, Yisong Su, Huanhuan Ma, Xiaoyan Yu, Fangyu Lei, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2023. Menatqa: A new dataset for testing the temporal comprehension and reasoning abilities of large language models.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report.

Yifan Zeng. 2024. Histolens: An llm-powered framework for multi-layered analysis of historical texts – a case application of yantie lun. *arXiv preprint arXiv:2411.09978*.

Bowen Zhao, Zander Brumbaugh, Yizhong Wang, Hannaneh Hajishirzi, and Noah A. Smith. 2024. Set the clock: Temporal alignment of pretrained language models.

Mengyu Zheng, Hanting Chen, Tianyu Guo, Chong Zhu, Binfan Zheng, Chang Xu, and Yunhe Wang. 2024. Enhancing large language models through adaptive tokenizers. In *Proc. NeurIPS*.

Zhejian Zhou, Jiayu Wang, Dahua Lin, and Kai Chen. 2024. Scaling behavior for large language models regarding numeral systems: An example using pythia.

A Appendix

Implementation details of evaluation. The evaluation pipeline is implemented in Python and supports asynchronous API requests with retry logic, as well as multiprocessing to handle thousands of examples efficiently. After collecting GPT-4o's label for each instance, we map `CORRECT/INCORRECT NOT ATTEMPTED` to categorical scores A, B, and

C. We then compute three core metrics: overall accuracy (proportion of A scores), given-attempted accuracy (A over A+B), and the F1 score, defined as the harmonic mean of overall and given-attempted accuracy. Results are reported both globally and stratified by task split (Context-based, Format Switching, Date Arithmetic) and by temporal category (Past, Near Past, Present, Future).

Date ambiguities. We explicitly enumerate all valid variants in the gold label set for each example to handle multiple correct answers arising from date-format ambiguities. This ensures that any prediction matching one of these variants is marked correct, avoiding penalisation for format differences.

Synthetic benchmark construction for linear probing. We construct a suite of synthetic true–false benchmarks to isolate temporal reasoning across different reference frames. For the DATES_PAST, DATES_PRESENT, and DATES_FUTURE datasets, we sample 1,000 date–date pairs each, drawing calendar dates uniformly from the appropriate range and rendering them in two randomly chosen, distinct formatting patterns (`Ymd` vs `d/m/Y`). Exactly half of each set are "YES" examples (identical dates under different formats), which are our positive examples, and half are "NO" (different dates), which are our negative examples. All three datasets are balanced, shuffled, and split into equal positive and negative subsets to ensure fair probing.

Models	**Past**	**Near Past**	**Present**	**Future**
GPT-4o-mini	61.66	67.93	70.51	58.23
OLMo-2-7B	49.45	62.35	73.56	43.45
Qwen2.5 14B	58.97	64.80	67.22	55.69
Qwen2.5 7B	51.41	55.98	57.98	48.55
Qwen2.5 3B	46.50	50.25	51.98	43.91
LLama3.1 8B	45.28	48.82	50.48	42.76
Qwen2.5 1.5B	42.99	46.16	47.69	40.60
Qwen2.5 0.5B	39.15	41.68	43.00	36.98
OLMo-2-1B	36.07	38.09	40.49	34.07
LLama3.2 3B	36.48	38.57	39.74	34.46

Table 5: Model accuracy on context-based questions across four data splits over time.

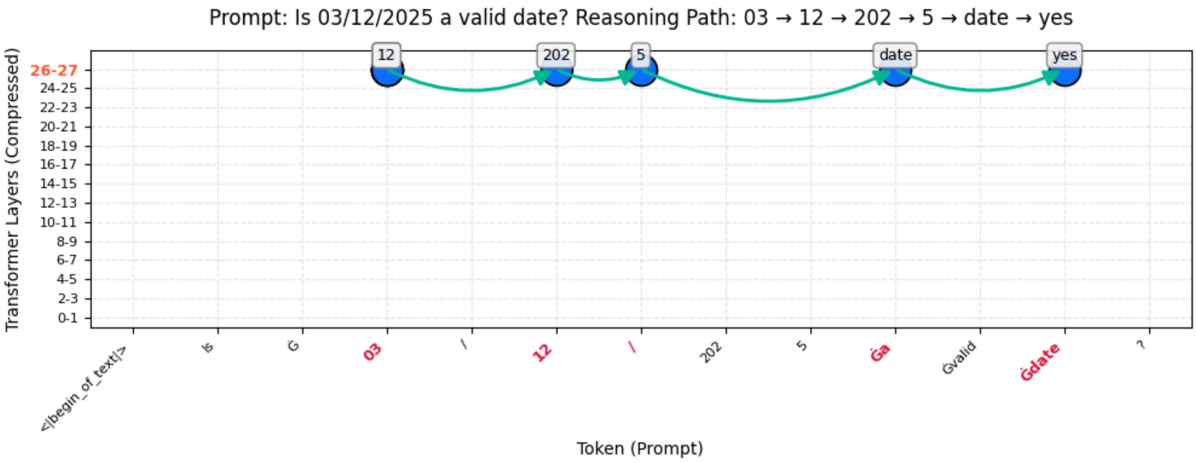


Figure 7: Causal-tracing of the “03/12/2025 is a valid date” judgment.

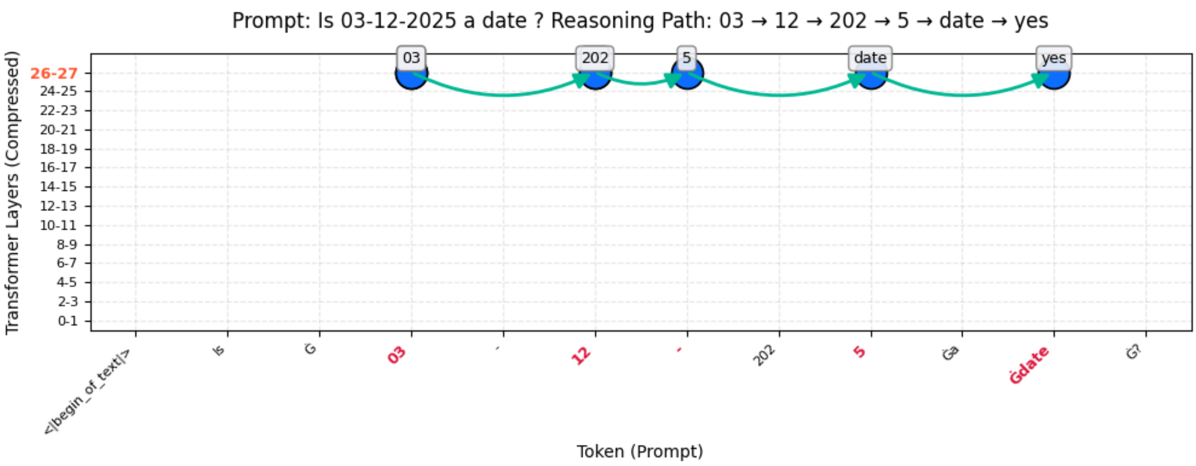


Figure 8: Causal-tracing of the “03-12-2025 is a valid date” judgment.

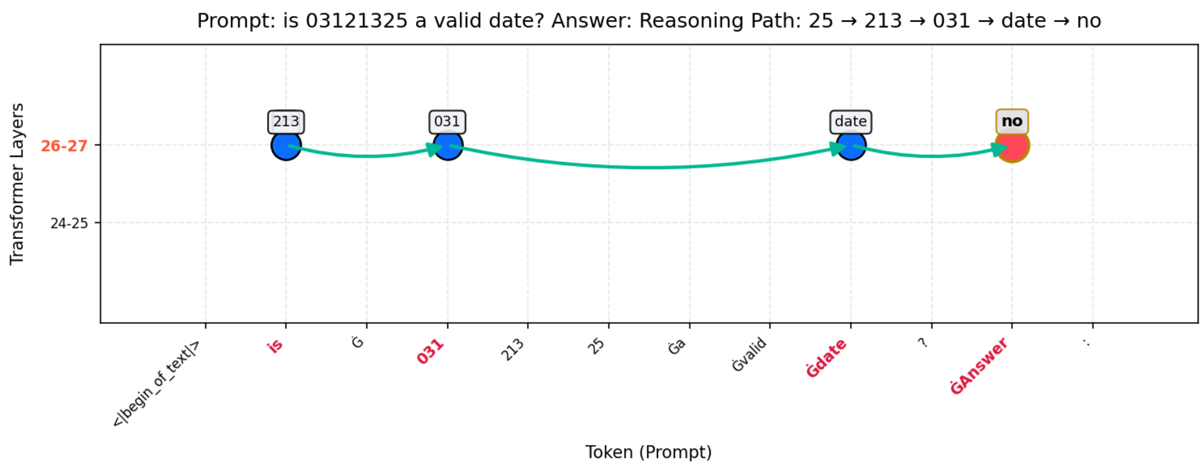


Figure 9: Causal-tracing of the “03121325 is a valid date” (Date in past) judgment.

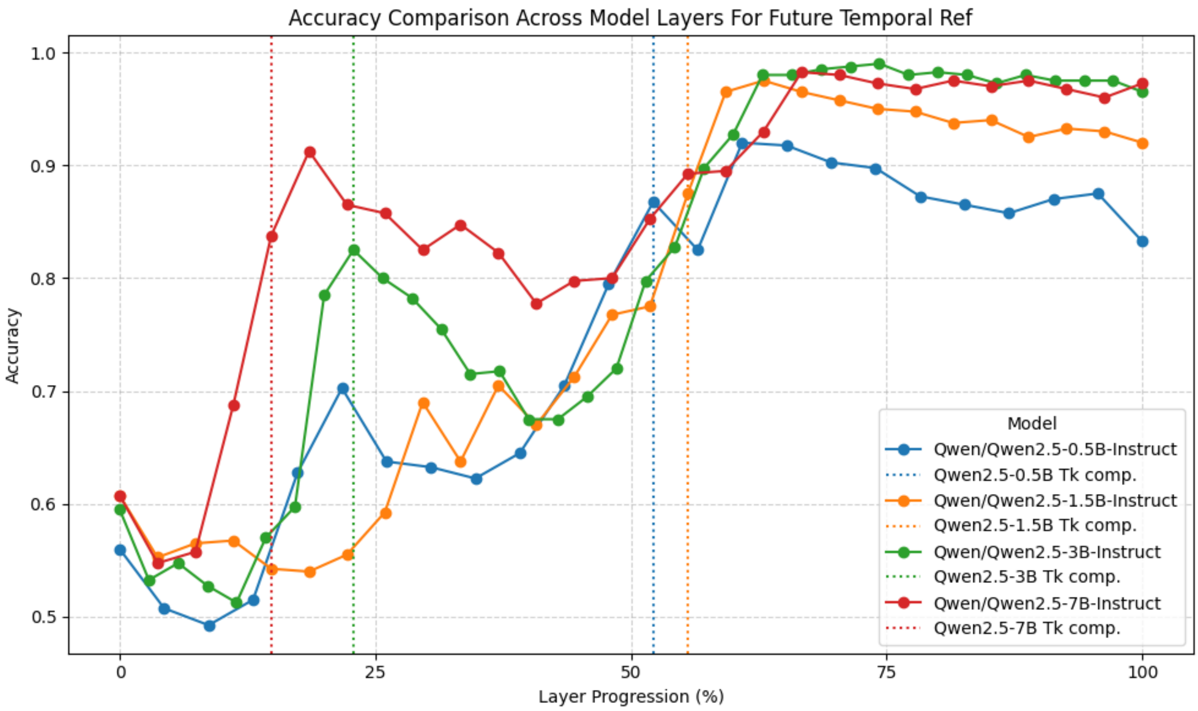


Figure 10: Layer-wise accuracies in the Future period

Model	DD-MM-YYYY	DD/MM/YYYY	YYYY/MM/DD	DDMMYYYY	MMDDYYYY	YYYYMMDD	Avg.
OLMo	64.70	64.56	65.35	52.35	54.56	50.41	58.65
Llama 3	50.31	50.89	53.45	38.45	40.24	34.56	44.65
GPT-4o	68.51	71.23	69.24	61.23	62.34	64.98	66.25
Qwen	64.49	62.35	73.56	46.50	50.25	51.98	58.19
Gemma	58.90	58.97	64.80	47.22	46.50	50.25	54.44
Phi	47.23	46.07	48.09	39.15	41.68	43.00	44.20

Table 6: Model accuracy on context-based questions across date formats.