TISER-based Multimodal Temporal Reasoning via RAG

Daniele Catalano
Politecnico di Torino
s349472
s349472@studenti.polito.it

Ramadan Mehmetaj
Politecnico di Torino
s346213
s346213@studenti.polito.it

Francesco Dal Cero
Politecnico di Torino
s342631
s342631@studenti.polito.it

Samuele Caruso
Politecnico di Torino
s349506
s349506@studenti.polito.it

***Abstract*—This work presents a multimodal framework for structured temporal reasoning within a Retrieval-Augmented Generation (RAG) setting. The task targets questions requiring reasoning over event ordering, overlap, and duration expressed in natural-language contexts. Textual temporal segments are automatically parsed into structured interval representations and rendered into synthetic timeline visualizations (Gantt, scatter-based intervals, and line sequences), producing aligned question–image–answer triples.**

Retrieval and generation are explicitly decoupled. A multimodal retriever ranks candidate timeline charts given a textual query, and retrieval quality is evaluated independently using Recall@k. The top-ranked chart is then provided as visual evidence to a vision-language model for temporally constrained answer generation.

End-to-end reasoning is measured using Exact Match (EM) and F1 across multiple benchmarks. Fine-tuning improves performance over the baseline, particularly on complex tasks, while retrieval grounding yields consistent gains and approaches fine-tuned accuracy. These results show that explicit visual timeline grounding enhances temporal consistency and reliability in multimodal RAG systems.

I. Introduction

Temporal reasoning—understanding [2] time, duration, and event sequencing—remains a challenging capability for Large Language Models (LLMs). While modern models show strong linguistic competence, they often struggle with long-horizon chronological consistency. This limitation is particularly critical in Retrieval-Augmented Generation (RAG), where retrieved fragments must be integrated into a coherent temporal structure.

Structured reasoning frameworks such as Timeline Self-Reflection (TISER) [7] partially address this issue. TISER employs a multi-stage process based on textual timeline construction, iterative self-reflection, and test-time scaling, improving temporal consistency over standard chain-of-thought prompting.

However, TISER operates purely in text and human temporal reasoning is often spatial and diagrammatic. Moreover, current Vision-Language Models (VLMs) lack explicit grounding in structured temporal layouts.

To address this gap, we extend TISER into a multimodal framework , where an initial preprocessing pipeline extracts and normalizes temporal entities into structured intervals and generates visual timelines. These visual artifacts provide explicit grounding signals that reduce hallucination and promote interval-level consistency. Figure 1 illustrates the overall architecture, integrating dataset construction, multimodal indexing within a RAG module, and vision-language inference. The full implementation and experimental setup are publicly available on the project repository (link project github).

The main contributions are:

- **Multimodal Temporal Pipeline:** An automated system for extracting and normalizing temporal information to generate structured visual charts.
- **Multimodal RAG Framework:** A retrieval-aware architecture indexing timeline visualizations as structured visual documents.
- **Fine-Tuning and Evaluation:** A VLM fine-tuning [3] strategy leveraging multimodal grounding to improve temporal reasoning over textual baselines.

II. Extensions

A. Context-Aware Temporal Figures Construction

The pre-processing pipeline transforms textual temporal prompts into aligned structured and visual representations for multimodal retrieval and reasoning. The procedure operates offline and is applied consistently across all dataset splits, including `TimeQA-Easy`, `TimeQA-Hard`, `L2`, `L3`, and `TGQA`.

The temporal context is isolated and treated as the primary source of chronological structure. Rule-based patterns extract event descriptions and their temporal anchors.

Visualization is performed at the context level rather than per individual sample, in fact questions sharing the same temporal context are grouped via a unique identifier, and a single structured temporal table and corresponding timeline are generated and reused across associated questions. This design eliminates redundant image generation and ensures consistency among samples derived from the same narrative. A total of 2k unique context-level graphs are used for fine-tuning, while an additional 300 graphs are reserved for test-time inference evaluation.[1]

The parser supports heterogeneous formats, including year ranges (“YYYY – YYYY”), interval expressions (“X from YYYY to YYYY”), and TGQA-style annotations such as

[1]Datasets available for training at https://huggingface.co/datasets/Dancat/MultiModal_TISER_train-dataset. and test https://huggingface.co/datasets/Dancat/MultiModal_TISER_test_only-dataset

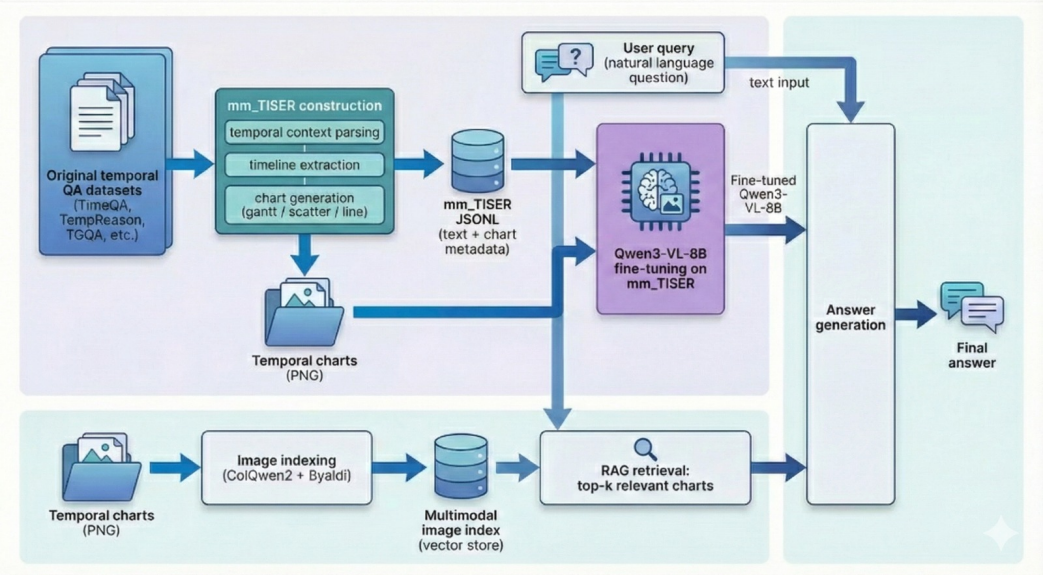

Fig. 1. Overview of the proposed multimodal temporal reasoning architecture. The system extracts temporal entities, generates structured visual timelines, and integrates retrieval with vision-language reasoning for grounded inference.

"event starts at YYYY" or "event ends at YYYY". Extracted events are normalized into a *multi-granular timeline* representation, where annual resolution serves as the primary temporal scale while month-level information is preserved when available through decimal conversion. When only partial boundaries are present, events are represented consistently as either intervals or point events to maintain schema coherence.

Each context is converted into a structured temporal table containing event descriptions and temporal spans, which serves as the basis for visualization.

During image generation, the chart type is randomly selected among *Gantt charts*, *scatter-based* interval plots, and *line-plots*, more details on Appendix A. Visual attributes such as color palette, background, grid style, and font family are also randomized to prevent stylistic overfitting and encourage reliance on temporal structure rather than graphical patterns.

Each visualization is aligned with its associated questions, answers, and metadata, producing a multimodal dataset of question–image–answer triples.

The full charts generation pseudo-code is illustrated in Fig. 2, and it can be summerized into three stages:

Stage I: Contextual Temporal Extraction. The system isolates the temporal context embedded in the prompt and parses it directly, without relying on external metadata. Rule-based patterns extract event descriptions and temporal boundaries, supporting heterogeneous formats across the datasets. All values are normalized to a year-level numeric scale, optionally incorporating month precision. When only one boundary is available, the event is represented as a point event to maintain schema consistency.

Stage II: Structured Timeline Construction. Extracted events are assembled into tuples $(e_i, y_i^{start}, y_i^{end})$, where e_i denotes the i-th event and y_i^{start} and y_i^{end} represent its starting and ending year, respectively, incorporating month-level information when available. For punctual events, $y_i^{start} = y_i^{end}$. Chronological validity is enforced and duplicate entries are removed. The structured table is then rendered into a visual timeline. As it was said before, the chart type and the visual styling are randomized in order to have a more robust learning signal.

Stage III: Multimodal Alignment. Each visualization is aligned with its question, answer, dataset identifier, and meta-

Algorithm 1 Multimodal Dataset Creation with Timeline Charts

Input: Temporal prompt dataset $\mathcal{D}$
Output: New multimodal dataset $\mathcal{D}_{MM}$ with visual timelines

// Stage I: Contextual Temporal Extraction
for each instance $d \in \mathcal{D}$ **do**
  Isolate contextual temporal segment from $d.prompt$
  Apply rule-based parsing to extract events and temporal anchors
  Normalize temporal spans into structured representations
  // Stage II: Timeline Chart Generation
  Construct temporal table $\{(e_i, t_i^{start}, t_i^{end})\}$
  Sort events chronologically
  Generate timeline chart (Gantt / Scatter / Line) with randomized styling
  // Stage III: Multimodal Dataset Construction
  Associate generated chart with question, answer, and metadata
  Store multimodal sample in $\mathcal{D}_{MM}$
end for

Fig. 2. Pipeline for constructing a new multimodal dataset by transforming textual temporal prompts into structured representations and corresponding timeline charts for fine-tuning and retrieval-based reasoning.

data such as chart type and number of events. The final sample includes the image path and structured temporal attributes. This alignment enables explicit chronological grounding in retrieval-based settings, reducing reliance on textual pattern matching.

The pipeline enforces consistent multi-granular timeline while preserving the contextual temporal information of the original prompts, improving robustness and interpretability in temporal question answering tasks.

B. RAG Construction and Retrieval Framework

Retrieval-Augmented Generation (RAG) [6] is adopted as the downstream evaluation setting to assess whether multimodal temporal representations improve context-aware validation and interval-based reasoning. In temporal question answering, correct inference depends on retrieving the appropriate timeline context prior to generation. Retrieval and reasoning are therefore explicitly decoupled, enabling controlled evaluation of both components.

The pipeline consists of three stages: multimodal indexing, text-to-image retrieval, and post-retrieval visual reasoning.

Index Construction. All generated charts visualizations are indexed using a multimodal retriever implemented through `RAGMultiModalModel`. The backbone relies on ColQwen2, a late-interaction vision-language architecture inspired by ColPali (Figure 3) and ColBERT-style multi-vector retrieval. Text queries and images are encoded into sets of embeddings, and relevance is computed through patch-level similarity aggregation rather than global embedding comparison.

Each timeline visualization is treated as a self-contained visual document. Indexing is performed at the context level: each unique temporal narrative corresponds to a single stored visualization, even if multiple questions reference it.

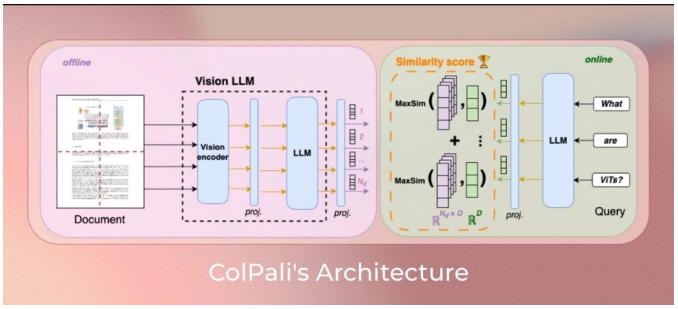

Fig. 3. Late-interaction multimodal retrieval architecture inspired by ColPali. Text queries and images are encoded into multi-vector representations and matched via token-level similarity aggregation.

Text-to-Image Retrieval. At inference time, the textual query is encoded and matched against indexed visual embeddings using late-interaction scoring. The top-k most relevant timeline images are retrieved. Retrieval quality is evaluated using Recall@k, measuring whether the gold timeline appears within the top-k candidates. Results are reported both globally and per visualization type.

RAG + Vision-Language Inference. The top-1 retrieved visualization is provided to a vision-language model for answer generation. The model receives both the timeline image and the textual query and is instructed to reason over event intervals and temporal coverage constraints. End-to-end performance is measured using Exact Match (EM) and F1, capturing the combined effect of retrieval precision and structured visual reasoning.

The late-interaction multi-vector design preserves localized alignment signals between textual queries and visual elements, which is particularly beneficial in structured synthetic charts where interval boundaries and ordering cues serve as critical retrieval anchors.

C. Fine-tuning Strategy

To effectively integrate visual temporal grounding into the reasoning process, the **Qwen3-VL-8B-Instruct** [4] model, a state-of-the-art Vision-Language Model, was fine-tuned to better interpret and reason over multi-granular timeline charts. The fine-tuning procedure was specifically designed to enhance the model's capability to align textual queries with structured temporal visual elements, while preserving its original instruction-following and multimodal reasoning abilities.

The resulting fine-tuned model is publicly available[2].

QLoRA (Quantized Low-Rank Adaptation), a parameter-efficient fine-tuning technique, was employed to mitigate the substantial memory requirements of the 8B parameter architecture. The model was trained using a Supervised Fine-Tuning (SFT) objective, where the input sequence interleaves the generated timeline image with the textual query. The training objective reinforces a structured inference process: the model is required to analyze the visual context to verify temporal constraints before generating the final answer. Detailed hyperparameters and experimental configurations are provided in Appendix B.

D. Prompt Design and Inference Procedure

During inference, both the system-level instruction and the user prompt were reformulated to enforce temporally consistent reasoning over timeline charts. The objective was to constrain the model to perform explicit interval validation rather than relying on surface-level correlations.

At the system level, a structured Chain-of-Thought (CoT) protocol was implemented, requiring sequential visual reasoning, the extraction of relevant temporal events, and explicit verification that the queried date or range is fully covered by an event interval.

To ensure deterministic evaluation, intermediate reasoning stages are constrained within dedicated internal tags, while the final output must be strictly enclosed in `<answer>` tags. Only the content inside this tag is considered during scoring.

[2]Fine-tuned model available at https://huggingface.co/Dancat/MM_Tiser_Qwen3_VL_FT_v2.

The user prompt further enforces the process by specifying that answers must be derived from the order, overlap, and duration of events shown in the chart. When no event fully satisfies the queried temporal constraint, the model is required to output "Unknown".

The full text of the system instruction and the complete prompt template are provided in Appendix C, while an illustrative inference example is presented in Section D.

III. Results

First evaluation: The retrieval component of the RAG pipeline is assessed prior to downstream question answering. Table I reports Recall@k over 300 held-out multimodal test instances, measuring how often the ground-truth chart appears within the top-k retrieved results, thereby isolating ranking performance from generative reasoning.

Second evaluation: End-to-end reasoning is evaluated using Exact Match (EM) and F1 across five temporal benchmarks. Table II compares baseline and fine-tuned [5] models over the same 300 test instances (60 per dataset), assessing whether supervised adaptation improves temporal reasoning beyond retrieval quality.

Third evaluation: The effect of Retrieval-Augmented Generation (RAG) on reasoning accuracy [1] is examined by measuring EM and F1 after retrieval phase. Results are reported in Table III.

TABLE I
RETRIEVAL PERFORMANCE ON THE TISER TEST SET (RECALL@K IN %).

Chart Type	Recall@1	Recall@3	Recall@5
Gantt	54.9	59.8	71.9
Scatter	66.4	68.6	72.1
Line	67.9	82.1	83.3
Global (All)	**63.7**	**69.7**	**75.0**

Table I shows a global Recall@1 of 63.7%, increasing to 75.0% at Recall@5, indicating that most errors stem from ranking rather than retrieval failure. Line charts achieve the highest accuracy, followed by scatter plots, while Gantt charts remain more challenging, suggesting that continuous temporal representations are easier to retrieve than interval-dense layouts.

TABLE II
BASELINE AND FINETUNED PERFORMANCE ACROSS TEMPORAL REASONING BENCHMARKS (N=60 PER DATASET).

Dataset	EM (Base)	F1 (Base)	EM (FT)	F1 (FT)
TimeQA (hard)	53.33	0.571	60.00	0.623
TempReason (L3)	45.00	0.521	50.00	0.558
TGQA	41.67	0.602	46.67	0.758
TempReason (L2)	41.67	0.497	43.33	0.514
TimeQA (easy)	76.67	0.771	76.67	0.772
Macro Avg.	**51.67**	**0.592**	**55.33**	**0.645**

Table II indicates that fine-tuning improves macro-averaged EM from 51.67% to 55.33% and F1 from 0.592 to 0.645. Gains are most pronounced on reasoning-intensive datasets such as TimeQA (hard), TempReason (L3), and TGQA, while performance on TimeQA (easy) remains stable. This confirms that supervised adaptation enhances temporal reasoning without degrading simpler cases.

TABLE III
POST-RAG END-TO-END PERFORMANCE ON THE MULTIMODAL TEMPORAL REASONING TEST SET (N=300).

Setting	EM	F1
Post-RAG (Qwen)	53.33	0.636

Table III reports end-to-end reasoning performance after integrating RAG. The Post-RAG configuration achieves 53.33% EM and 0.636 F1 over the held-out test set. Compared to the pre-RAG baseline (51.67% EM; 0.592 F1), retrieval grounding yields a consistent improvement, particularly in F1 (+0.044), indicating enhanced semantic alignment and more accurate interval validation.

Although the fine-tuned model remains slightly superior in exact match accuracy (55.33% EM; 0.645 F1), the Post-RAG setting substantially narrows the performance gap without requiring supervised adaptation. These results demonstrate that retrieval module provides a meaningful contribution to temporal reasoning consistency and improves robustness in multimodal interval-based inference.

IV. Conclusion and Limitations

This work presented a multimodal temporal reasoning framework that converts textual temporal contexts into structured visual timelines within a Retrieval-Augmented Generation (RAG) pipeline. By leveraging question answering in explicit chronological representations, the approach reduces reliance on superficial textual correlations and promotes interval-based reasoning. Results demonstrate consistent improvements over the baseline and performance close to supervised fine-tuning, highlighting the effectiveness of retrieval-based grounding.

However, several limitations remain. Performance depends on retrieval accuracy: ranking errors or partial mismatches can propagate to the reasoning stage and limit downstream results. Although Recall@k indicates that most failures stem from ranking rather than complete retrieval misses, imperfect grounding still constrains overall accuracy.

Additionally, automatically generated timeline charts may introduce visual artifacts. In dense scenarios with overlapping events, compressed labels or occlusions can reduce clarity and affect multimodal reasoning. Sensitivity to resolution and scaling further impacts robustness.

Finally, temporal normalization may simplify complex event relations, and the combined retrieval–generation architecture increases computational cost compared to purely generative models.

The proposed multimodal RAG framework offers an interpretable and scalable approach to structured temporal reasoning, showing that explicit visual grounding enhances interval-level inference despite practical constraints.

REFERENCES

[1] A. Author et al., "Evaluation Metrics for Retrieval-Augmented Generation," arXiv:2504.14891, 2025. [Online]. Available: https://arxiv.org/abs/2504.14891

[2] A. Author et al., "Temporal Reasoning in Large Language Models: A Comprehensive Study," arXiv:2504.05258, 2025. [Online]. Available: https://arxiv.org/abs/2504.05258

[3] Hugging Face, "Fine-tuning Vision-Language Models with TRL," 2024. [Online]. Available: https://huggingface.co/learn/cookbook/fine_tuning_vlm_trl

[4] Qwen Team, "Qwen3-VL-8B Technical Report," arXiv:2511.21631, 2025. [Online]. Available: https://arxiv.org/abs/2511.21631

[5] A. Author et al., "A Study on Efficient Hyperparameter Tuning for Multimodal Large Language Models," arXiv:2406.05130, 2024. [Online]. Available: https://arxiv.org/abs/2406.05130

[6] A. Author et al., "REAL-MM-RAG: A Real-World Multi-Modal Retrieval Benchmark," arXiv:2502.12342, 2025. [Online]. Available: https://arxiv.org/abs/2502.12342

[7] A. Bazaga, R. Blloshmi, B. Byrne, and A. de Gispert, "Learning to Reason Over Time: Timeline Self-Reflection for Improved Temporal Reasoning in Language Models," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 28014–28033. [Online]. Available: https://aclanthology.org/2025.acl-long.1358.pdf

APPENDIX

A. Visual Representations and Design Rationale

To enforce temporal reasoning in visual modalities, the pipeline dynamically selects the most appropriate chart type based on the topological properties of the temporal entities (e.g., point-wise events versus continuous durations) and the narrative structure. Three distinct visualization strategies are employed: *Gantt Charts*, *Scatter Plots (Interval Sequences)*, and *Line Charts (Temporal Sequences)*.

Gantt Charts (Duration-Centric) For narratives necessitating the analysis of time spans, overlapping periods, or concurrent events (e.g., monarchical reigns, war durations), Gantt-style charts are utilized.

- **Visual Structure:** Temporal entities are represented as horizontal bars extending from a start year (t_{start}) to an end year (t_{end}). Bars are stacked vertically to prevent occlusion.
- **Rationale:** Textual models frequently struggle with "interval algebra"—specifically, determining intersection or inclusion between two distinct timeframes. By mapping duration to geometric length and concurrency to vertical alignment, Gantt charts allow the resolution of temporal overlap problems via visual inspection rather than through complex symbolic arithmetic.

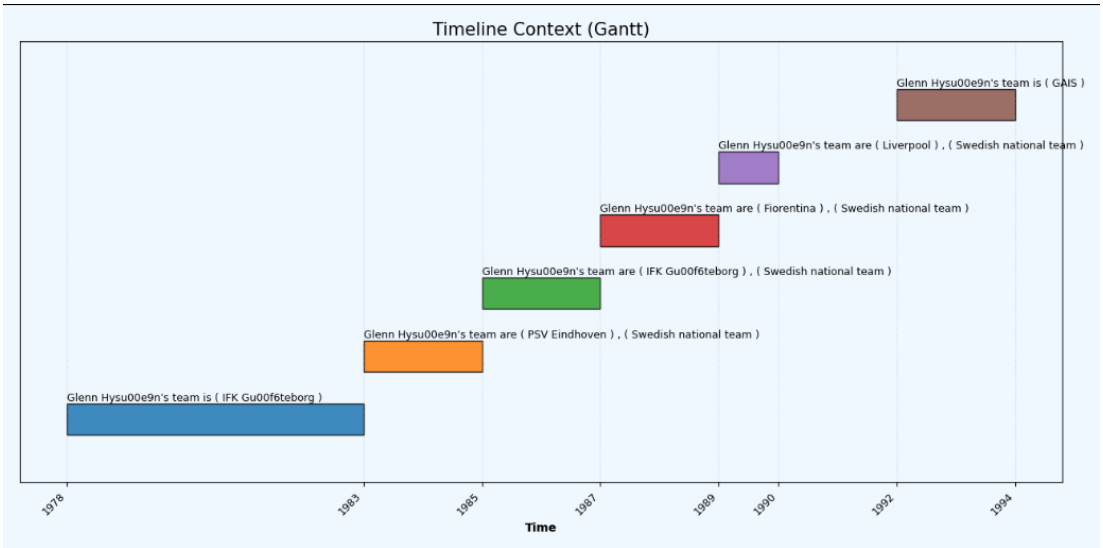

Fig. 4. Example of a *Gantt Chart* generated for duration-centric narratives. Horizontal bars denote temporal intervals (e.g., lifespans, political terms), enabling the model to visually deduce start/end points and resolve queries regarding overlapping periods or concurrency.

Scatter Plots (Interval Sequence)

When the narrative consists of a high density of discrete events where the specific distribution or clustering is more relevant than strict sequential connectivity, Scatter Plots are employed.

- **Visual Structure:** Events are plotted as discrete markers (nodes) along the time axis without connecting edges. The vertical axis is utilized to separate distinct categorical entities or to reduce visual overlap (jittering).
- **Rationale:** In scenarios characterized by non-linear narratives or disjoint event clusters, explicit connecting lines can introduce false causality or visual clutter. Scatter plots preserve the temporal sparsity of the data, facilitating a focus on the density and isolation of specific events without the bias of a continuous narrative line.

Line Charts (Temporal Sequence)

For narratives implying a strong causal flow or a strictly chronological sequence of distinct events (e.g., a sequence of steps, major milestones in a biography), Line Charts are selected.

- **Visual Structure:** Events are represented as nodes connected by edges, forming a continuous path along the temporal axis ($t \to t+1$).
- **Rationale:** The connecting lines explicitly visualize the progression and rate of change between events. This structure reinforces the concept of "narrative flow," aiding in the tracking of long-horizon dependencies and the understanding of the relative "velocity" of the narrative (e.g., steep slopes indicate rapid successions of events).

Design Optimization for VLMs

Across all three chart types, strict stylistic constraints are imposed to accommodate the optical character recognition (OCR) and encoder behavior of the Qwen2-VL architecture:

Fig. 5. Example of a *Scatter Plot* (Interval Sequence) used for high-density, discrete event data. This representation avoids visual clutter by plotting events as independent nodes, facilitating the identification of temporal clustering and density without implying a strict continuous connectivity or false causality.

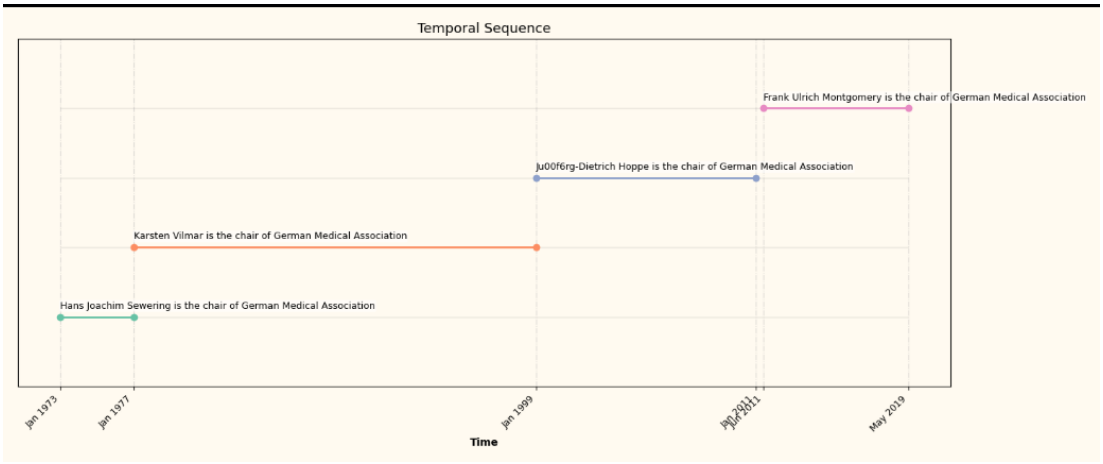

Fig. 6. Example of a *Line Chart* (Temporal Sequence) representing a linear narrative flow. The connected path emphasizes the chronological progression and relative timing between successive events, supporting sequential reasoning tasks and the analysis of narrative velocity.

- **High-Contrast Palettes:** A distinct color cycle is utilized to ensure boundaries between adjacent elements are sharp, preventing tokenization errors in the visual encoder.
- **Minimalist Formatting:** Grid lines, background shading, and decorative axes are removed to maximize the signal-to-noise ratio for the attention heads.
- **Label Orientation:** Text labels are constrained to horizontal orientation or minimal angles ($< 30°$) to ensure readability, as vertical text often degrades performance in current VLM encoders.

B. Implementation Details

Hardware and Environment

All experiments were conducted on a high-performance computing node equipped with a single **NVIDIA A100 GPU (80GB VRAM)**, an Intel Xeon Platinum 8470 Processor, and 128GB of system RAM. This configuration enabled the efficient processing of high-resolution chart images within the multimodal context window.

Training Data and Splits

Fine-tuning was performed using a dataset of 2,077 unique temporal context graphs derived from the TISER corpus. These graphs were expanded into multiple question–image–answer triples used for training. For evaluation, a strictly held-out test set of 300 instances was employed, balanced across five benchmark subsets (TimeQA-Easy, TimeQA-Hard, TempReason L2, TempReason L3, and TGQA), with 60 triples per subset.

Hyperparameters and Configuration
The QLoRA method was utilized by injecting Low-Rank Adapters into the linear projection layers of the self-attention mechanism and feed-forward networks (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`).

The specific training configuration was as follows:

- **LoRA Parameters:** Rank $r = 16$, Alpha $\alpha = 32$, Dropout $= 0.05$.
- **Optimization:** AdamW (8-bit) optimizer with a learning rate of 1×10^{-4} and a cosine learning rate scheduler.
- **Batch Size:** Per-device batch size of 2 with gradient accumulation steps of 4, resulting in an effective batch size of 8.
- **Loss Function:** Standard cross-entropy loss on the target tokens.

C. Prompt Templates

To ensure full reproducibility, the verbatim system and user prompt templates employed during both fine-tuning and inference are reported below. The prompting strategy enforces a structured multi-stage reasoning procedure designed for temporal chart understanding.

System Message
The system message enforces a four-stage structured reasoning pipeline (Reasoning → Timeline → Reflection → Answer). The model is required to strictly follow the XML-style tagging format shown below.

SYSTEM_PROMPT

```
You are an AI assistant that uses a Chain of Thought (CoT) approach with reflection
to answer queries about charts.
Follow these steps:
Step 1. Reason through the visual data step by step within the <reasoning> tags.
Step 2. Given your previous reasoning, identify relevant temporal events in the given
context for answering the given question within <timeline> tags. Assume relations in the
context are unidirectional.
Step 3. Reflect on your reasoning and the timeline to check for any errors or improvements
within the <reflection> tags.
Step 4. Make any necessary adjustments based on your reflection. If there is additional
reasoning required, go back to Step 1 (reason through the visual data step-by-step),
otherwise move to the next step (Step 5).
Step 5. Provide your final, concise answer within the <answer> tags.
If the answer is a number, just output the number, nothing else.
Otherwise output the entity or event, without any additional comments.
Important: The <reasoning>, <reflection> and <timeline> sections are for your internal
    reasoning
process. All the reflection and the timeline have to be contained inside the thinking
    section.
Do not use enumerations or lists when writing, use plain text instead such as paragraphs.
The response to the query must be entirely contained within the <answer> tags.
Use the following format for your response:
<reasoning>
[Your step-by-step reasoning goes here. This is your internal thought process.] <timeline>
[Relevant temporal events for answering the given question.]</timeline> <reflection>
[Your reflection on your reasoning, checking for errors or improvements]</reflection>
[Any adjustments to your thinking based on your reflection]</reasoning> <answer>
[Your final, concise answer to the query.] </answer>
When answering, always follow these rules:
- Use the chart to reason about the order, overlap, and duration of events, and answer
    exactly what is asked in the question.
- Identify the event or interval that actually covers the requested date or date range on
    the timeline.
- If the requested period is a range (e.g. 2006-2007), the correct event must cover the
    whole range, not just its start or end.
- If no event covers the requested date or the whole requested range, answer 'Unknown' (or
    the event labeled as Unknown in the chart).
- Never pick an event only because it is the last or the most recent one. Always check
    whether its interval includes the queried date(s).
```

User Message Template

The user prompt integrates the visual input and the textual query following the format required by the Qwen2-VL multimodal architecture.

USER_PROMPT

```
Question: {user_query}
Temporal context: The provided chart contains the temporal context for this question.
Important: Use the chart to reason about the order, overlap and duration of events, and
answer exactly what is asked in the question.
When the question asks about a specific date or date range, identify the event whose
interval actually includes that date or fully covers that range.
If the chart does not provide enough information to answer, answer Unknown.
```

D. Demo example

Figure 7 presents a representative inference example using the fine-tuned Qwen3-VL within the proposed multimodal RAG framework. Given a temporal question, the retriever selects the most relevant timeline chart, which is provided to the vision-language model together with the query and structured system prompt.

The model follows a structured reasoning protocol, generating intermediate representations within `<reasoning>`, `<timeline>`, and `<reflection>` tags, verifying temporal coverage constraints, and extracting the final prediction strictly from the `<answer>` tag. This example illustrates how the integration of retrieval and structured inference supports accurate temporal reasoning in a multimodal setting.

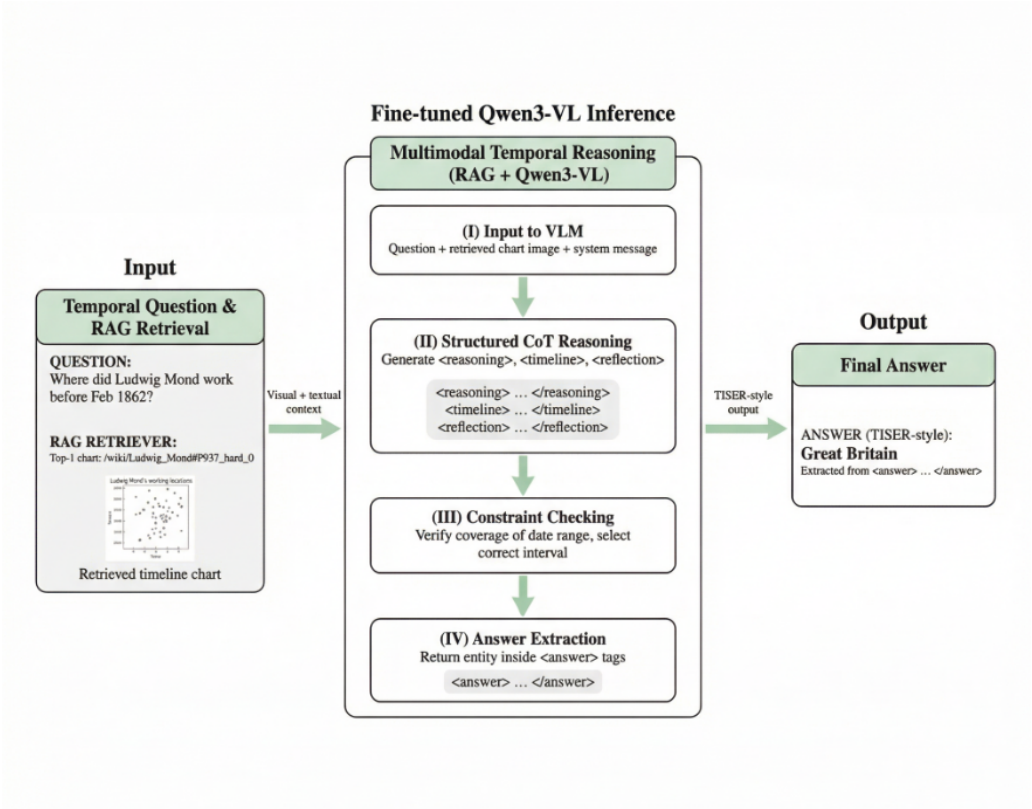

Fig. 7. End-to-end inference example using the fine-tuned Qwen3-VL within the proposed multimodal RAG framework. The retriever selects the most relevant timeline chart, which is provided together with the query to the VLM. The model follows a structured reasoning protocol and extracts the final prediction from the `<answer>` tag.