Time-Series Retrieval for Grounding Multimodal Language Models in Remaining Useful Life Prediction

Valeriu Dimidov and Raphaël Frank

Interdisciplinary Centre for Security, Reliability and Trust (SnT)
University of Luxembourg
29 Avenue J.F. Kennedy L-1855, Luxembourg
firstname.lastname@uni.lu

ABSTRACT

Large language models (LLMs) and agentic AI systems are increasingly being explored for domain-specific maintenance and prognostics tasks, raising the question of whether they can effectively support prognostics and health management (PHM). In this paper, we investigate remaining useful life (RUL) estimation with multimodal large language models (MLLMs) grounded through time-series retrieval. We propose a framework in which historically similar degradation segments are retrieved from the training set and, together with the test trajectory, transformed into a visual comparison artifact that is processed by the MLLM through a structured multimodal prompt. The approach is evaluated on the FD001 partition of the C-MAPSS benchmark under repeated experiments comparing retrieval-based inference against a non-retrieval baseline based on random reference selection. The results show that time-series retrieval consistently improves MLLM-based RUL prediction across the evaluated models, yielding lower error and more stable performance. At the same time, the magnitude of the benefit depends on model capacity, indicating that retrieval is most effective when the underlying MLLM is able to exploit the retrieved evidence. Overall, the study shows that time-series RAG is a promising mechanism for improving multimodal prognostic reasoning, while also highlighting the current limitations of MLLM-based RUL estimation in practical PHM settings.

1. INTRODUCTION

Predictive Maintenance (PdM) aims to reduce downtime, maintenance costs, and operational risks by anticipating failures before they occur. A key task in this context is Remaining Useful Life (RUL) estimation, which provides an estimate of the time left before a component reaches failure. Accurate RUL estimation supports maintenance planning, resource allocation, and risk-aware decision making in Prognostics and Health Management (PHM).

Conventional RUL estimation methods are typically based on statistical degradation models, physics-informed approaches, or data-driven architectures trained on sensor time series. Although these methods can achieve strong predictive performance, they usually operate only on numerical data and provide limited support for reasoning over external contextual knowledge.

Recent advances in large language models (LLMs) and multimodal large language models (MLLMs) create an opportunity to revisit RUL estimation from a different perspective (Zhang, Chowdhury, Gupta, & Shang, 2024; Yin et al., 2024). Instead of relying only on a learned numerical predictor, an MLLM can be prompted with visual and textual evidence describing the target trajectory. Retrieval-augmented generation (RAG) can further enrich this input by selecting historical degradation segments that are similar to the query trajectory, following the broader idea of grounding language-model generation with retrieved external evidence (Lewis et al., 2020). However, in the RUL setting, the effect of such retrieval is not obvious: relevant references may ground the model on useful degradation evidence, whereas poorly matched references may introduce noise or bias the prediction.

Therefore, this paper studies whether time-series retrieval improves MLLM-based RUL estimation. We propose a framework in which similar degradation segments are retrieved from historical run-to-failure trajectories and combined with the query trajectory into a visual comparison artifact. The same multimodal prompt structure is then used to compare retrieval-based inference with a non-retrieval baseline based on random reference selection on the FD001 partition of the C-MAPSS benchmark (Saxena, Goebel, Simon, & Eklund, 2008; DeCastro, Litt, & Frederick, 2008).

The main contributions of this study are:

- A multimodal framework that reformulates RUL estima-

tion as an evidence-grounded visual reasoning task for MLLMs.

- A time-series RAG mechanism that retrieves historical degradation segments and presents them to the MLLM as trajectory-level reference evidence.
- A controlled experimental comparison between retrieval-based and random-reference inference on the FD001 partition of the C-MAPSS benchmark.
- An analysis of the benefits, limitations, and failure modes associated with retrieval augmentation in the RUL estimation setting.

The remainder of the paper is organized as follows: Section 2 reviews related work, Section 3 presents the methodology, Section 4 reports the experimental results, Section 5 discusses the main findings and limitations, and Section 6 concludes the paper.

2. RELATED WORK

LLMs have started to attract attention in time-series analysis. Recent survey studies indicate that this line of research mainly follows several directions, including direct prompting, time-series quantization into token-like representations, alignment of numerical sequences with language-model spaces, the use of visual representations as an intermediate reasoning interface, and the integration of LLMs with external tools (Zhang et al., 2024). In parallel, the rapid development of multimodal large language models (MLLMs) has shown that language-centered models can reason jointly over heterogeneous modalities such as text, images and audio, which is particularly relevant when sensor trajectories are converted into plots or textual summaries before inference (Yin et al., 2024).

In the specific context of PHM and RUL estimation, the use of LLMs and MLLMs is still recent but growing. One contribution introduced an LLM-based regression framework for turbofan RUL prediction and reported competitive results together with promising transfer-learning behavior (Y. Chen & Liu, 2024). Another study explored a pre-trained LLM-based approach for aircraft-engine RUL prediction on C-MAPSS (Tan, Yang, Zhu, & Wang, 2025). More recently, a multimodal framework was proposed to jointly exploit temporal signals, frequency-domain images, and textual domain knowledge for industrial time-series analysis, and it was evaluated on the four standard C-MAPSS subsets (Wang et al., 2026). At a broader PHM level, another work proposed a language-model-based framework intended to support multiple maintenance-related tasks within a unified setting (Ren et al., 2025).

Another relevant direction concerns the integration of external knowledge into LLM-based PHM pipelines. RAG was introduced to combine parametric language models with explicit non-parametric memory, thereby improving grounding and updateability (Lewis et al., 2020). In predictive maintenance, recent studies have argued that retrieval and knowledge augmentation can help LLM-based systems access maintenance records, technical documentation, and operational procedures more effectively (Jiang & Hu, 2025).

Despite these advances, only a limited number of studies explicitly investigate the influence of retrieval augmentation on MLLM-based RUL estimation in a controlled setting. Existing studies mainly focus either on adapting LLMs and MLLMs to prognostics tasks or on using textual retrieval for broader PHM support. Consequently, the effect of time-series RAG on RUL prediction quality remains insufficiently studied (Y. Chen & Liu, 2024; Tan et al., 2025; Wang et al., 2026; Jiang & Hu, 2025; Hafsi, 2025; Kirubanandan, 2025).

3. METHODOLOGY

This section describes the problem setup, proposed model pipeline, and experimental configuration.

3.1. Problem Definition

We consider a set of n equipment units, denoted by $U = \{U_0, U_1, \ldots, U_{n-1}\}$, monitored over time through onboard measurements. Each unit U_i is represented by a multivariate time series $X_i \in \mathbb{R}^{T_i \times M}$, where $T_i \in \mathbb{N}$ denotes the number of observed cycles and M is the number of monitored variables. The corresponding sequence of RUL targets is denoted by $R_i \in \mathbb{N}_0^{T_i}$.

Given the dataset

$$\mathcal{D} = \{(X_i, R_i)\}_{i=0}^{n-1}, \tag{1}$$

the objective is to learn a mapping function

$$f_{TS} : \bigcup_{T \in \mathbb{N}} \mathbb{R}^{T \times M} \to \mathbb{N}_0 \tag{2}$$

such that, for a given time series X_i, the function f_{TS} predicts the RUL of its last observation.

In this work, the mapping function f_{TS} is approximated using a multimodal large language model. Since MLLMs are designed to process multimodal inputs such as images and text, each time series X_i is transformed into a multimodal representation Z_i, which may include visual depictions of sensor trajectories and textual contextual information. The prediction function can therefore be expressed as

$$f_{MM} : \mathcal{Z} \to \mathbb{N}_0, \tag{3}$$

where $\mathcal{Z}$ denotes the space of multimodal representations derived from the original time series. The goal remains the estimation of the RUL of the last observation of each unit.

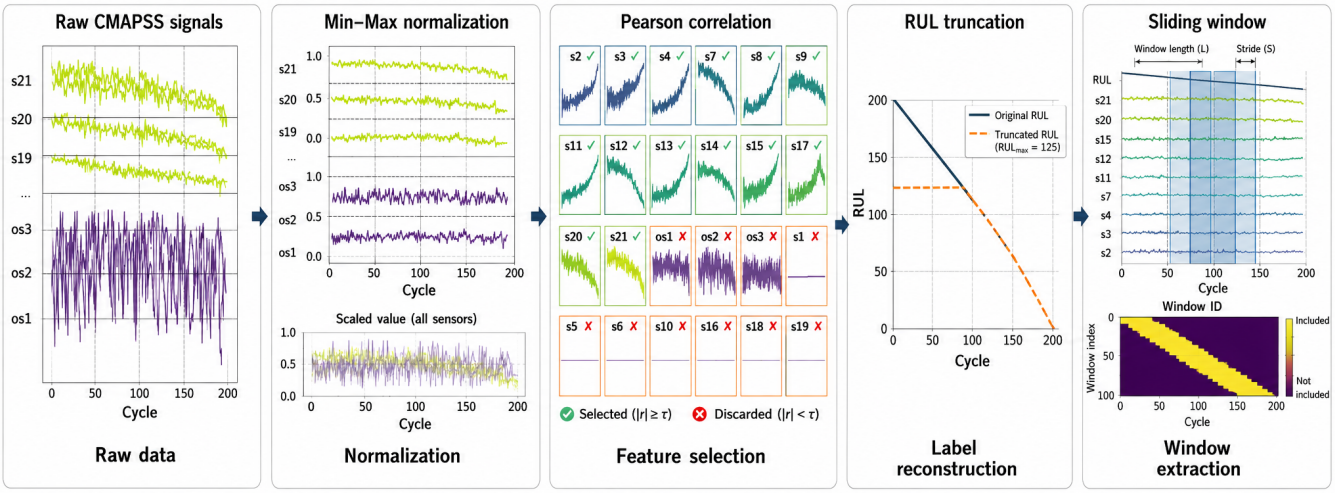


Figure 1. Preprocessing workflow applied to the FD001 partition of the C-MAPSS dataset.

Table 1. Overview of the main pipeline modules according to their inputs, transformations, and outputs.

Module	Input	Transformation	Output
Segment encoder	Normalized sliding-window segment	Encode with an LSTM autoencoder.	Fixed-dimensional embedding.
Retrieval memory builder	Training segments and RUL labels	Sample by RUL bin and index embeddings.	Balanced vector database.
Query retriever	Last observed test segment	Encode query and perform k-nearest-neighbor search.	Top-k historical references.
Trajectory comparison builder	Query trajectory and retrieved references	Align trajectories and generate a multisensor plot.	Trajectory-comparison image.
Prompt composer	Comparison image and task instructions	Insert visual evidence into a structured prompt.	Multimodal prompt for RUL estimation.

3.2. Dataset

This study uses the C-MAPSS benchmark for the empirical evaluation of RUL estimation. C-MAPSS is a widely used simulated turbofan-engine degradation dataset in which each trajectory corresponds to one equipment unit monitored over successive operating cycles through three operational setting variables and multiple sensor measurements. The training split contains complete run-to-failure trajectories, whereas the test split contains truncated trajectories that end before failure. Accordingly, the task is to estimate the RUL at the last observed cycle of each test unit (Saxena et al., 2008; DeCastro et al., 2008).

Owing to economic, computational, and time constraints, the experimental campaign reported in this paper is restricted to the FD001 partition. This design choice enables a focused and controlled analysis within a feasible experimental budget, while a broader evaluation on the remaining partitions is deferred to future work.

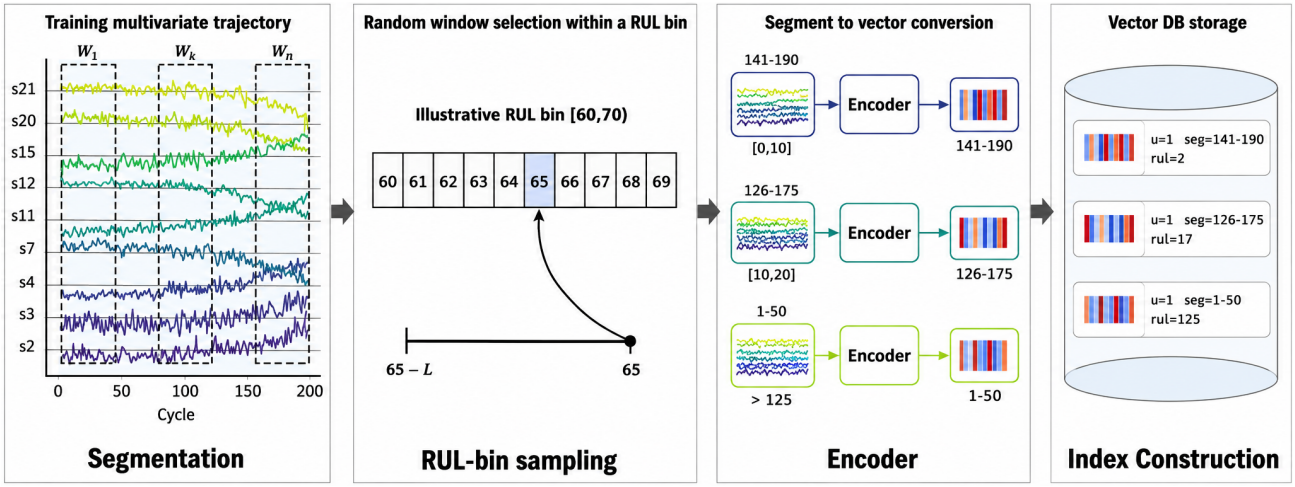
For further details on the simulator and dataset construction, the reader is referred to (Saxena et al., 2008; DeCastro et al., 2008).

3.3. Preprocessing

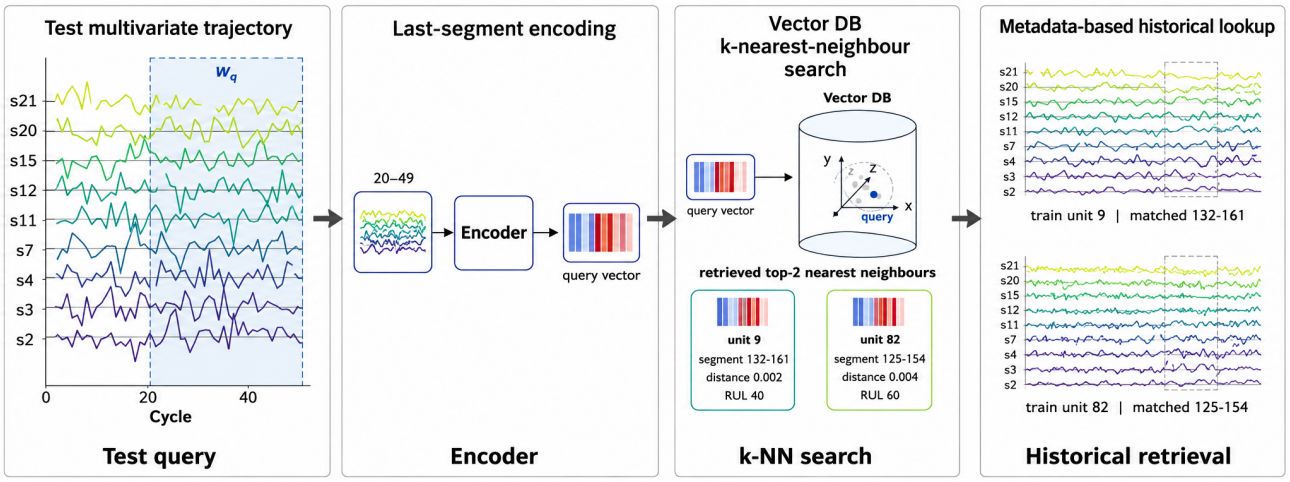
The preprocessing pipeline transforms the raw C-MAPSS trajectories into standardized fixed-length samples suitable for both retrieval and multimodal inference, as illustrated in Figure 1. The raw trajectories are first grouped by equipment unit and ordered by cycle. Next, min–max normalization is applied independently to all sensor channels using statistics computed from the training set. This transformation reduces scale discrepancies across variables and yields a more comparable representation of the degradation behavior across units.

After normalization, feature selection based on Pearson correlation is applied to reduce redundancy and eliminate non-informative measurements. Only the variables selected through this process are kept for the subsequent stages.

The cycle-level RUL targets are then reconstructed from the distance to failure, and a truncation threshold of 125 cycles is applied. Finally, each normalized trajectory is segmented through a sliding-window procedure. Given a window length L, the method extracts fixed-length temporal segments, and each segment is associated with the RUL value of its last cycle. This step converts variable-length degradation trajec-



(a) RAG builder. Training trajectories are segmented, sampled by RUL bin, encoded, and indexed in the vector database.



(b) Query retriever. The last segment of the test trajectory is encoded and used to retrieve the nearest indexed train segments.

Figure 2. Overview of the time-series RAG mechanism: (a) offline indexing of encoded training segments; (b) online retrieval using the encoded test segment.

tories into standardized samples that can be used consistently by the retrieval module and by the multimodal inference pipeline.

3.4. Time-Series RAG Mechanism

The proposed time-series RAG mechanism is composed of three main modules: the encoder module, the retrieval memory builder, and the query retriever. Figure 2 summarizes the overall process, distinguishing between the offline construction of the retrieval memory and the online retrieval of historical references for a test unit.

3.4.1. Encoder Module

The encoder module transforms each normalized sliding-window segment into a compact vector representation used for similarity-based retrieval. Its purpose is to represent the recent temporal behavior of a unit, including sensor levels and local degradation trends, in a fixed-dimensional embedding space.

Let S_j denote a normalized segment produced by the preprocessing step introduced in Section 3.3. The encoder defines a mapping

$$z_j = \phi(S_j) \in \mathbb{R}^d$$

where z_j is the embedding of segment S_j and d is the embedding dimension.

In this work, the encoder is obtained through an LSTM-based autoencoding scheme trained exclusively on windows extracted from the training trajectories. The LSTM encoder processes the multivariate sensor segment and maps it to a latent embedding. During training, a lightweight decoder is attached to this embedding and optimized to reconstruct the final sensor state of the segment.

After training, the decoder is discarded and only the encoder, together with the embedding projection, is retained. The main architectural and training settings of the encoder are summarized in Table 2.

Table 2. Configuration of the learned LSTM encoder used to generate retrieval embeddings.

Aspect	Configuration
Input window length	$L = 30$ cycles
Input variables	Selected sensor features
Encoder type	LSTM encoder
Number of recurrent layers	1
Embedding layer	Linear projection
Embedding dimension	$d = 64$
Hidden dimension	128
Decoder type	Feed-forward head
Training objective	Final-step reconstruction
Loss function	Mean squared error
Optimizer	AdamW
Training epochs	200
Batch size	128
Learning rate	10^{-3}
Weight decay	10^{-5}
Embedding normalization	L2 normalization

3.4.2. Retrieval Memory Builder

The retrieval memory builder constructs the non-parametric memory used by the proposed time-series RAG mechanism. This module corresponds to the offline stage shown in Figure 2(a). It operates only on the training partition, in order to avoid information leakage, and transforms historical degradation data into a searchable collection of representative examples.

Window-to-Embedding Transformation. The module is built from the windowed training segments produced by the preprocessing step described in Section 3.3. Each segment is associated with the RUL value of its last cycle. The trained encoder described in Section 3.4.1 is then used to transform each selected segment into a fixed-dimensional embedding.

RUL-Balanced Retrieval Memory Construction. To reduce redundancy and improve coverage across degradation stages, the training segments are grouped according to their RUL values, and a subset of representative segments is sampled from each RUL bin.

The resulting retrieval memory can be represented as

$$\mathcal{C} = \{(z_j, r_j, m_j)\}_{j=1}^{N}$$

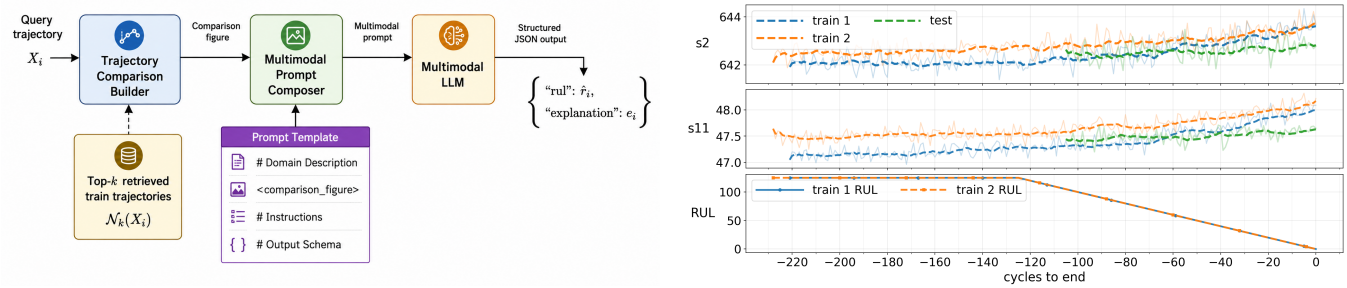
where z_j is the segment embedding defined by the encoder module, r_j is the associated RUL value, and m_j contains metadata such as the source unit and cycle position. The embeddings and metadata are then stored in a vector database, forming a searchable memory of historical degradation examples.

3.4.3. Query Retriever

The query retriever is the online module responsible for selecting the historical references used to ground the MLLM prediction. This module corresponds to the online stage shown in Figure 2(b). For each test unit, only the last observed segment is used as the query, since the objective is to estimate the RUL at the most recent cycle.

The query segment is encoded using the same encoder adopted during memory construction. This ensures that both training and test segments are represented in the same embedding space. Given the last observed query segment Q_i, its embedding is obtained as

$$z_i^{(q)} = \phi(Q_i)$$



(a) Multimodal prompt-generation workflow.

(b) Example of the trajectory-comparison image.

Figure 3. Multimodal prompt generation and example comparison artifact used for MLLM inference.

A nearest-neighbor search is then performed in the retrieval memory to identify the k most similar historical segments:

$$\mathcal{N}_k(Q_i) = \text{top-}k_{j\in\{1,\dots,N\}}\ \text{sim}\left(z_i^{(q)}, z_j\right)$$

where $\text{sim}(\cdot,\cdot)$ denotes the similarity measure used by the retrieval engine.

The retrieved neighbors provide historical references whose recent sensor behavior is close to that of the test unit. Their metadata are used to recover the corresponding training trajectories, cycle positions, and RUL values. These references are then used in the next stage to generate the visual comparison artifact for multimodal prompting.

3.5. Multimodal Prompt Generation and MLLM Inference

The multimodal prompt-generation stage is composed of three main modules: the trajectory comparison builder, the multimodal prompt composer, and the structured output parser. Figure 3 summarizes the overall process, showing how the query trajectory and the retrieved historical references are transformed into a visual comparison artifact, inserted into a structured multimodal prompt, and processed by the MLLM to produce an RUL estimate and a textual explanation.

3.5.1. Trajectory Comparison Builder

The Trajectory Comparison Builder transforms the query trajectory and the retrieved historical references into a trajectory-comparison image. This image contrasts the recent behavior of the test unit with similar training trajectories, allowing the MLLM to inspect sensor levels, temporal trends, and degradation-stage alignment in a compact visual form.

As shown in Figure 3a, this visual artifact is generated before prompt composition and serves as the main grounding evidence for the subsequent MLLM-based RUL prediction.

Visual encoding choices. The visual artifact is designed to make the degradation comparison readable within the constraints of a paper figure and an MLLM input. The query trajectory and the retrieved references are plotted on a common temporal axis, so that their recent behavior can be compared directly. The selected sensor channels are shown as separate trajectories, allowing the model to evaluate whether the similarity between the query and the references is consistent across multiple variables.

The example reported in Figure 3b includes two selected sensors and the corresponding RUL profile for readability. In the actual inference pipeline, the comparison artifact incorporates the sensor variables selected during preprocessing, as described in Section 3.3. The RUL information associated with the retrieved training references is used to contextualize their degradation stage, while the RUL of the test unit remains the target to be estimated.

These visual encoding choices are intended to provide the MLLM with evidence about three main aspects: the current degradation level of the query unit, its recent temporal trend, and its similarity to historical run-to-failure examples.

3.5.2. Multimodal Prompt Composer

The Multimodal Prompt Composer combines the trajectory-comparison image with a textual prompt template, as shown in Figure 3a. The resulting prompt provides the MLLM with visual evidence, a description of the RUL estimation task, instructions for comparing the query trajectory with the retrieved references, and an explicit output schema.

This structure ensures that all evaluated MLLMs receive a consistent multimodal input. The system prompt, instruction prompt, and user prompt template used in this study are reported in the Appendix, Figure 5.

3.5.3. Structured Output Parsing

For each query unit, the MLLM is required to return a structured response containing both the numerical RUL estimate and a short textual explanation. The expected output has the

form

$$y_i = \left\{ \begin{array}{l} \text{“rul”} : \hat{r}_i, \\ \text{“explanation”} : e_i \end{array} \right\},$$

where $\hat{r}_i \in \mathbb{N}_0$ denotes the predicted RUL of the last observation of the query trajectory, and e_i is the explanation generated by the model.

The predicted RUL value is used for quantitative evaluation, whereas the explanation provides qualitative information about the reasoning process followed by the MLLM. In this way, the model is used not only as a regressor, but also as a reasoning component that can justify its prediction with reference to the visible degradation evidence. A representative explanation returned by the model is provided in Section 5.

3.6. Experimental Protocol

The experimental protocol compares two reference-selection strategies for multimodal RUL estimation: *random* trajectory selection and *RAG-based* trajectory selection. Let $\mathcal{M}$ denote the set of evaluated MLLMs. The experiment is repeated $N = 10$ times to account for variability induced by the MLLM inference parameters and by the non-deterministic construction of the RAG component.

At each repetition n, a repetition-specific environment

$$E^{(n)} = \left(\theta^{(n)}, \mathcal{C}^{(n)}, \mathcal{I}^{(n)}\right)$$

is instantiated, where $\theta^{(n)}$ denotes the sampled inference parameters, $\mathcal{C}^{(n)}$ is the reference corpus derived from the training set, and $\mathcal{I}^{(n)}$ is the corresponding retrieval index. For each reference-selection strategy, the prompts are generated once under the current environment and then used to query all MLLMs in $\mathcal{M}$. This produces a paired evaluation setting in which, within each repetition, all models are assessed using the same prepared prompts.

Input: Train set $\mathcal{D}_{train}$, test set $\mathcal{D}_{test}$, window length L, number of references k, prompt template T, set of MLLMs $\mathcal{M}$, parameter search space Θ, repetitions N
Output: Predictions and scores for all strategies, MLLMs, and repetitions

for $n \leftarrow 1$ **to** N **do**
  $E^{(n)} \leftarrow$ BUILDENVIRONMENT$(\mathcal{D}_{train}, L, \Theta)$;
  foreach $s \in \{\texttt{random}, \texttt{rag}\}$ **do**
    $\mathcal{P}^{(n,s)} \leftarrow$ PROMPTS$(\mathcal{D}_{test}, E^{(n)}, s, k, T)$;
    foreach $m \in \mathcal{M}$ **do**
      $\hat{\mathcal{Y}}^{(n,s,m)} \leftarrow$ CALLLLM$(\mathcal{P}^{(n,s)}, m, E^{(n)})$;
      $M^{(n,s,m)} \leftarrow$ SCORE$(\hat{\mathcal{Y}}^{(n,s,m)})$;
    end
  end
end

Algorithm 1: Experimental protocol used to compare random and RAG-based reference selection across multiple MLLMs over repeated non-deterministic environments.

Algorithm 1 summarizes the full evaluation procedure. BUILDENVIRONMENT samples the inference parameters, constructs the reference corpus, and builds the retrieval index. PROMPTS extracts the last observed test segments, selects the references according to the considered strategy, generates the comparison figures, and instantiates the prompt template. Finally, CALLLLM queries each MLLM, and SCORE evaluates the resulting RUL predictions. The scores are aggregated across repetitions to compare random and retrieval-based reference selection.

Evaluated MLLMs. The experimental evaluation considers three Gemini-based multimodal large language models, namely *Gemini 3.1 Flash-Lite*, *Gemini 3 Flash*, and *Gemini 3.1 Pro*. These models were selected to cover different levels of model capacity and computational cost. In particular, Flash-Lite represents a lighter and more efficient variant, Flash provides an intermediate configuration, and Pro corresponds to a more capable model intended for more demanding reasoning tasks. This selection allows us to analyze whether the effect of retrieval augmentation is consistent across models with different capability–efficiency trade-offs.

4. RESULTS

Figure 4 reports the predictive performance of the three evaluated Gemini MLLMs under the two considered inference settings, namely random multimodal inference (*MLLM+Random*) and retrieval-augmented multimodal inference (*MLLM+RAG*). Across all evaluated models, the inclusion of time-series retrieval leads to a consistent improvement in predictive quality, yielding lower error and more stable behavior than random reference selection. This

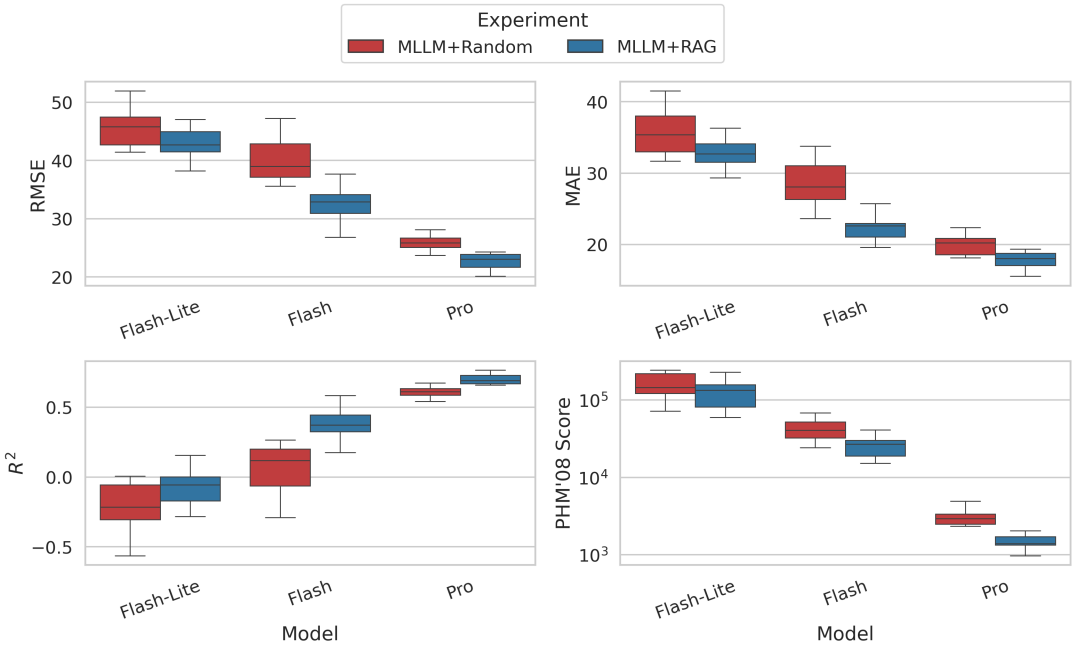


Figure 4. Predictive performance over 10 repetitions for the three evaluated Gemini MLLMs under the two inference settings, *MLLM+Random* and *MLLM+RAG*.

Table 3. FD001 test-set comparison between baseline methods and Gemini-based RAG variants.

	Baselines			Gemini RAG		
Metric	**Mean**	**Random predictor**	**LSTM (Z. Chen et al., 2021)**	**Flash-Lite**	**Flash**	**Pro**
RMSE	41.98 ± 0.00	57.19 ± 2.49	14.54	43.12 ± 2.72	32.85 ± 3.30	22.69 ± 1.44
MAE	36.07 ± 0.00	47.53 ± 2.66	–	32.91 ± 2.19	22.47 ± 1.99	17.91 ± 1.24
R^2	-0.021 ± 0.000	-0.897 ± 0.165	–	-0.080 ± 0.135	0.370 ± 0.125	0.701 ± 0.037
PHM'08 Score	20302 ± 0	181531 ± 74903	322.44	127441 ± 66610	27712 ± 15830	1488 ± 323

trend indicates that the retrieved historical trajectories provide useful grounding evidence for the MLLM during RUL estimation.

Among the proposed configurations, the strongest results are obtained by the *Pro + RAG* setting, while *Flash + RAG* also shows clear gains over its non-retrieval counterpart. The *Flash-Lite* variant benefits less from retrieval, suggesting that the usefulness of the retrieved information depends not only on the retrieval mechanism itself, but also on the reasoning capacity of the underlying multimodal model. Overall, the results support the claim that similarity-based reference selection is preferable to arbitrary reference sampling when MLLMs are used for RUL estimation.

The comparison reported in Table 3 further helps position the proposed approach with respect to the literature. On the one hand, the proposed RAG-based variants outperform simple baselines and improve consistently over the corresponding random-reference MLLM setting. On the other hand, the best proposed configuration does not yet surpass the strongest task-specific and more sophisticated deep learning architecture included in the comparison.

A second relevant observation concerns performance dispersion across the repeated experiments. For the *Flash* and *Pro* variants, the RAG-based setting tends to produce more compact distributions than the non-retrieval baseline, especially for the main error metrics. This suggests that retrieval not only improves the central tendency of the predictions, but also reduces sensitivity to arbitrary reference selection. In summary, the results indicate that time-series retrieval is a beneficial component within the proposed MLLM pipeline, even though the overall architecture remains simpler than the strongest predictive models reported in the literature.

5. DISCUSSION

The experimental results provide a positive answer to the main research question of this paper: in the considered FD001 setting, time-series retrieval improves MLLM-based RUL estimation. The observed gains suggest that retrieved training trajectories provide useful grounding evidence, enabling the model to compare the query with historically

similar degradation patterns rather than relying only on generic multimodal reasoning over the input artifact. In this sense, the proposed time-series RAG mechanism strengthens the link between the observed trajectory and previously seen run-to-failure behaviors.

Another relevant aspect concerns explainability. Unlike conventional regression baselines, the proposed framework can return not only a numerical RUL estimate but also a textual explanation of the rationale behind the prediction. Although this advantage does not compensate for the current performance gap with stronger specialized models, it remains valuable in PHM settings, where analysts may benefit from inspecting the prediction together with the retrieved historical references and the model's explanation. This explanatory capability is one of the motivations for exploring language-model-based approaches in prognostics.

To illustrate this qualitative aspect, Example 1 reports a representative explanation produced by the MLLM. The response shows that the model is able to relate the predicted RUL to observable properties of the query trajectory, such as the absence of visible degradation and the similarity with early-life train references. We also manually inspected a subset of the explanations and found them broadly consistent with the visual evidence provided in the trajectory-comparison figures. However, explanation quality was not evaluated systematically in this study.

Example 1. Representative explanation returned by the MLLM.

The TEST unit's sensor readings across all informative channels (such as s2, s3, s4, s7, s11, and s12) are completely flat and show no signs of degradation. The levels closely match the baseline values of the TRAIN references during their earliest recorded cycles, well before any degradation trends begin. Since there is no visible progression towards failure, the unit is clearly in its early life stage. Therefore, the estimated RUL is 125 cycles.

At the same time, the comparison with the literature must be interpreted carefully. The proposed approach does not yet match the predictive accuracy of the strongest architectures included in Table 3. However, this gap should be read in light of the purpose of the study. The main objective of this paper is not to introduce the most sophisticated possible multimodal architecture for RUL prediction, but rather to isolate and evaluate the influence of time-series RAG within a comparatively simple and controlled pipeline.

Several additional limitations should nevertheless be acknowledged. First, the evaluation is restricted to FD001, which is the simplest C-MAPSS subset. Therefore, the findings should not yet be generalized to more challenging settings involving multiple operating conditions, multiple fault modes, or previously unseen failure mechanisms. In particular, the proposed time-series RAG component can only retrieve degradation patterns that are represented in the indexed historical memory. If a failure mode has not occurred before, or is not sufficiently covered by the training trajectories, the system may retrieve only partially similar references and provide misleading grounding evidence to the MLLM. Second, the current study relies on a specific prompt design, a specific visual comparison artifact, and a single retrieval formulation, which may influence the quality of both the numerical predictions and the generated explanations. Third, although the experiment is repeated multiple times to account for variability, MLLM-based inference remains inherently non-deterministic and the proposed framework does not provide a formal guarantee of correctness. The repeated evaluation should therefore be interpreted as an empirical robustness assessment rather than as a correctness guarantee. Finally, the present analysis remains mainly descriptive and does not yet include systematic uncertainty estimation, novelty detection, or mechanisms for flagging cases in which the retrieved evidence is insufficient. Altogether, these limitations support interpreting the present work as a preliminary study on the role of retrieval in MLLM-based prognostics.

6. CONCLUSIONS

This paper presented a framework for studying RUL estimation with multimodal language models under retrieval augmentation. The proposed approach combines a time-series RAG module, which retrieves historically similar train segments, with a multimodal prompting pipeline that transforms the query and retrieved references into an input suitable for MLLM-based inference.

The empirical results on FD001 show that retrieval augmentation consistently improves the proposed MLLM pipeline relative to non-retrieval multimodal inference. Across all evaluated Gemini variants, the RAG-based setting yields better predictive performance and more stable behavior than random reference selection. These results support the main conclusion of the paper: time-series RAG has a positive influence on multimodal RUL estimation.

More broadly, the study highlights two promising aspects of language-model-based approaches for prognostics. First, they can benefit from explicit access to relevant historical trajectories rather than relying only on internal model reasoning. Second, they can naturally provide textual explanations alongside the numerical prediction, which may be valuable in human-centered PHM decision support settings.

Several directions can be explored in future work. First, the evaluation should be extended beyond FD001 to the more challenging C-MAPSS subsets and to additional predictive maintenance datasets. Second, the retrieval component could be improved through stronger embedding models, alternative similarity measures, and re-ranking strategies tailored to degradation trajectories. Third, future work should investi-

gate how time-series RAG interacts with more advanced multimodal architectures, including richer temporal backbones and more effective cross-modal fusion mechanisms. Finally, it would be valuable to study jointly the relationship between retrieval quality, predictive accuracy, and explanation quality in order to better understand when MLLM-based prognostic reasoning is most useful in practice.

ACKNOWLEDGMENT

This research was funded in whole, or in part, by the Luxembourg National Research Fund (FNR), grant reference BRIDGES/2022/IS/17270233. For the purpose of open access, and in fulfillment of the obligations arising from the grant agreement, the authors have applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

REFERENCES

Chen, Y., & Liu, C. (2024, October). *Remaining useful life prediction: A study on multidimensional industrial signal processing and efficient transfer learning based on large language models.* arXiv. doi: 10.48550/arXiv.2410.03134

Chen, Z., Wu, M., Zhao, R., Guretno, F., Yan, R., & Li, X. (2021, March). Machine remaining useful life prediction via an attention-based deep learning approach. *IEEE Transactions on Industrial Electronics*, *68*(3), 2521–2531. doi: 10.1109/TIE.2020.2972443

DeCastro, J., Litt, J., & Frederick, D. (2008). A modular aero-propulsion system simulation of a large commercial aircraft engine. In *44th AIAA/ASME/SAE/ASEE joint propulsion conference & exhibit.* Hartford, Connecticut: American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2008-4579

Hafsi, M. (2025). Potential of generative AI in knowledge-based predictive maintenance for aircraft engines. *Annual Conference of the PHM Society*, *17*(1). doi: 10.36001/phmconf.2025.v17i1.4352

Jiang, W., & Hu, F. (2025, August). Artificial intelligence agent-enabled predictive maintenance: Conceptual proposal and basic framework. *Computers*, *14*(8), 329. doi: 10.3390/computers14080329

Kirubanandan, R. (2025). Causal-aware LLM agents for PHM co-pilots: Health monitoring and intervention planning. *Annual Conference of the PHM Society*, *17*(1). doi: 10.36001/phmconf.2025.v17i1.4321

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in neural information processing systems* (Vol. 33, pp. 9459–9474). Curran Associates, Inc. doi: 10.48550/arXiv.2005.11401

Ren, J., Liu, X., Wang, T., Zhao, Z., Chen, X., Li, W., & Yan, R. (2025, November). PHM-GPT: A large language model for prognostics and health management. *Engineering*, S2095809925006745. doi: 10.1016/j.eng.2025.11.001

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008, October). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management* (pp. 1–9). Denver, CO, USA: IEEE. doi: 10.1109/PHM.2008.4711414

Tan, Q., Yang, L., Zhu, F., & Wang, Z. (2025). Pre-trained LLM-based remaining useful life prediction of aircraft engines. In *2025 15th international conference on quality, reliability, risk, maintenance, and safety engineering (QR2MSE) and 8th international conference on materials and reliability (ICMR)* (Vol. 2025, pp. 1016–1024). doi: 10.1049/icp.2025.3534

Wang, H., Li, Y., Zhu, Y., Yan, J., Ren, L., & Yang, L. T. (2026, March). *TS-MLLM: A multi-modal large language model-based framework for industrial time-series big data analysis.* arXiv. doi: 10.48550/arXiv.2603.07572

Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., & Chen, E. (2024, November). A survey on multimodal large language models. *National Science Review*, *11*(12), nwae403. doi: 10.1093/nsr/nwae403

Zhang, X., Chowdhury, R. R., Gupta, R. K., & Shang, J. (2024, May). *Large language models for time series: A survey.* arXiv. doi: 10.48550/arXiv.2402.01801

APPENDIX
Prompt Templates

System Prompt

You are an expert Remaining Useful Life (RUL) estimation analyst for industrial equipment monitored over time through multiple sensors.

Your task is to estimate the RUL of the TEST unit at the most recent cycle shown.

You will receive:

- a plot comparing a TEST time series against one or more TRAIN time series;
- optional contextual text describing how to interpret the trajectories.

Decision policy:

1. Estimate the TEST unit's latest RUL directly from the visual evidence.
2. Compare the TEST unit's recent sensor behavior against the TRAIN references.
3. Use visible evidence such as level, slope, curvature, and degradation stage to judge where the TEST unit lies in its degradation process.
4. Infer the RUL by assessing which TRAIN degradation stage the TEST unit most closely resembles, with emphasis on the most recent cycles.
5. Use agreement across multiple sensors more than any single noisy signal.
6. If the evidence is weak, noisy, or contradictory, return a cautious estimate consistent with the overall visible trend.

Constraints:

- Output MUST be valid JSON matching the schema.
- `rul` must be an integer in `[0,{clip_rul}]`.
- `explanation` must contain 2–5 sentences and must be grounded only in visible trends from the plot.
- Do NOT claim access to hidden metadata.

Instructions

- Focus on recent degradation trend and relative alignment to train trajectories.
- Give more weight to multi-sensor agreement than to isolated fluctuations.
- Infer degradation stage from visible evidence, not from any assumed external prediction.
- Prefer estimates that are visually well-supported by the TEST-to-TRAIN comparison.
- Keep uncertainty calibrated when the evidence is weak.

User Prompt Template

Task: estimate the Remaining Useful Life (RUL) of the TEST time series shown in this plot.

Problem description:

- Each unit is monitored over time through multiple sensor measurements.
- The TRAIN references show complete run-to-failure trajectories or long degradation trajectories from similar units.
- The TEST unit is only observed up to its current cycle, and your goal is to estimate how many cycles of useful life remain.
- Lower RUL means the unit appears closer to failure or to a late degradation stage.
- Higher RUL means the unit appears to be in an earlier degradation stage.

Important:

- Estimate the RUL directly from the plot.
- Do not assume any external model prediction is available.
- Base your estimate only on the visible sensor behavior and alignment with the TRAIN references.

What to do:

- Compare the TEST trajectory against the TRAIN references, especially in the most recent cycles.
- Focus on the recent degradation stage of the TEST unit.
- Use agreement across multiple sensors more than any single noisy signal.
- If the TEST unit appears close to late-life degradation patterns, predict a lower RUL.
- If the TEST unit appears to be at an earlier degradation stage, predict a higher RUL.
- If the evidence is mixed, provide a cautious estimate that best matches the overall visual pattern.

Return:

- `rul`: the estimated integer RUL for the most recent TEST cycle;
- `explanation`: a short explanation of how the visible sensor evidence supports the estimate.

Before generating the final answer, ensure that the predicted RUL value is compliant with all constraints and aligned with the explanation.

Figure 5. Prompt templates used for visual RUL estimation.