EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Human-Centered Maintenance Decision-Making in Smart Manufacturing Enhanced by English Technical Documents and Human-in-the-Loop Collaboration

Yanxia Quan[1,*]

[1]School of Foreign Languages, Huanghe Science and Technology University, Zhengzhou, 450061, Henan, China

Abstract

INTRODUCTION: Predictive maintenance is essential in human-centered smart manufacturing, yet existing methods often convert sensor data into fault diagnosis or RUL prediction without effectively transforming English technical documents into actionable maintenance constraints.
OBJECTIVES: This study aims to develop an interpretable human-in-the-loop maintenance decision-making framework that integrates equipment states, English technical clauses, and human feedback to support constraint-aware maintenance actions.
METHODS: An English technical document-enhanced framework is proposed, consisting of the English Document-Induced Constraint Learning (ELIC) module and the Human Feedback Loss Optimization (HFLO) module. ELIC models maintenance clauses from manuals, SOPs, and safety documents as state–document–action feasibility constraints. HFLO converts preference and risk feedback into optimization signals. A constraint-aware inference strategy combines task prediction scores, document feasibility scores, and feedback consistency scores to rank candidate actions.
RESULTS: Experiments on MetroPT-3 and XJTU-SY under the controlled protocol show that the proposed method achieves the lowest SP-CVR on both datasets and the highest F1 on MetroPT-3. On XJTU-SY, TCN–Transformer achieves a lower RMSE, while the human-in-the-loop baseline achieves a higher SP-FAS. Ablation results show that ELIC, HFLO, and constraint-aware inference affect different task and protocol-consistency metrics.
CONCLUSION: The proposed framework provides a traceable approach for jointly incorporating English technical clauses and feedback into maintenance action ranking under the constructed controlled protocol.

Keywords: human-centered smart manufacturing, predictive maintenance, English technical documents, human-in-the-loop, constraint learning

Received on 09 July 2026, accepted on 31 July 2026, published on 21 August 2026

doi: 10.4108/eetsis.13961

*Corresponding author. Email: quanyanxia83@163.com

1. Introduction

With the advancement of smart manufacturing, industrial sensing technologies, and predictive maintenance techniques, maintenance decision-making has become a central problem in human-centered smart manufacturing systems [1,2]. Existing predictive maintenance methods can effectively perform fault diagnosis, anomaly detection, and remaining useful life (RUL) estimation based on sensor data [3]. However, probabilistic fault predictions or RUL estimates are not directly equivalent to actionable maintenance decisions on the shop floor [4].

In real-world scenarios, maintenance decisions must integrate equipment manuals, operational procedures, safety instructions, and operator experience to determine whether a machine should continue operating, be monitored, warned,

inspected, repaired, or shut down. For cross-national industrial systems, such documentation is often provided in English [5]. Therefore, it is of practical significance to transform maintenance conditions, action constraints, and risk warnings embedded in English technical documents into computable decision constraints.

Existing research has explored industrial maintenance from multiple perspectives, including traditional maintenance strategies, data-driven predictive models, technical document-enhanced methods, and human-in-the-loop decision frameworks [6,7]. However, several limitations remain. Traditional methods struggle to dynamically integrate technical documentation with feedback signals. Data-driven approaches mainly focus on prediction accuracy, with limited attention to translating predictions into actionable maintenance decisions. Document-enhanced and retrieval-augmented methods typically treat documents as external context or explanatory material, without explicitly modeling action-level constraints [8]. Similarly, human feedback is often used for post-hoc evaluation or label correction, rather than being jointly optimized with document-derived constraints [9].

Consequently, the core challenge in maintenance decision-making is not only improving predictive accuracy, but also establishing a computable relationship among equipment states, English technical documentation, and feedback signals [10]. To address this issue, this paper proposes an English technical document-enhanced human-in-the-loop maintenance decision-making framework. The framework introduces an English document-induced constraint module that represents maintenance rules in manuals, SOPs, and safety documents as state–document–action feasibility signals. In addition, a feedback-aware optimization module incorporates preference and risk feedback into constraint-related learning objectives. Furthermore, constraint violation rate and feedback consistency are introduced to evaluate the utilization of both technical documents and human feedback. The contribution of this study is limited to their joint modeling under the constructed controlled maintenance protocol.

The main contributions of this paper are summarized as follows: (1) An English technical document-induced constraint learning mechanism is proposed to represent maintenance clauses as state–document–action feasibility signals, with clause-derived labels providing action-level supervision. (2) A feedback-aware optimization objective is developed to incorporate independently generated preference and risk feedback under document-induced feasibility states. (3) A constraint-aware maintenance action inference strategy is designed to integrate task prediction, document feasibility, and feedback consistency scores for final action ranking. (4) Experiments on MetroPT-3 and XJTU-SY evaluate task performance, the synthetic protocol constraint violation rate (SP-CVR), and the synthetic protocol feedback alignment score (SP-FAS). All claims are restricted to relative performance under the constructed controlled protocol rather than field-observed maintenance decisions or actual operator judgments.

2. Related Works

This chapter reviews research related to English technical document-enhanced human-in-the-loop maintenance decision-making, including traditional industrial maintenance, data-driven predictive maintenance, constraint-aware decision-making, technical document enhancement, human-in-the-loop and feedback learning, and the positioning of this work.

2.1 Traditional Industrial Maintenance and Decision-Making Methods

Traditional industrial maintenance decision-making mainly relies on manual inspection, periodic maintenance, expert experience, and rule bases. In complex industrial scenarios such as steel manufacturing, equipment often operates under high temperature, high load, and continuous operation conditions. Therefore, early studies mainly adopted threshold alarms, fault tree analysis, statistical process control, and manual rules to determine equipment states. These methods are suitable for scenarios where failure modes are clear and operating procedures are stable [11,12].

However, fixed thresholds and manual rules are easily affected by changes in operating conditions, equipment aging, and abnormal noise. Maintenance knowledge is also often distributed in equipment manuals, operating procedures, and safety instructions. Although traditional rule systems can encode part of the procedures, when facing document updates and feedback adjustments, they usually rely on manual maintenance and cannot form an optimizable decision signal. Therefore, traditional methods still have room for extension in jointly involving technical documents, equipment states, and feedback information in maintenance decision-making [13].

2.2 Data-Driven Predictive Maintenance and Prescriptive Maintenance Methods

Data-driven predictive maintenance has been widely used in complex industrial equipment management. Early methods mainly used logistic regression, support vector machines, random forests, and gradient boosting decision trees for fault classification or health state recognition. In recent years, LSTM, GRU, TCN, and Transformer models have been further used for sensor sequence modeling to capture degradation trends and predict remaining useful life [14,15].

These methods can improve equipment state prediction ability, but predictive maintenance mainly answers “whether the equipment is abnormal” or “what the remaining useful life is,” while maintenance decision-making still needs to answer “what action should be taken at the current time.” Prescriptive maintenance and decision optimization methods have begun to focus on transforming prediction results into maintenance strategies [16]. However, existing data-driven methods mainly use sensor data and historical labels as input, and still

lack explicit utilization of English technical document constraints and joint modeling of feedback signals [17,18].

2.3 Constraint-Aware Maintenance Decision-Making and Safety Optimization Methods

Constraint learning, safety optimization, and constraint-aware decision-making methods are commonly used for decision problems with safety boundaries, resource limitations, or policy feasibility requirements [19]. These methods typically enforce constraints through constraint losses, feasible region filtering, penalty terms, or Lagrangian optimization, enabling models to satisfy rule conditions while optimizing task objectives [20]. For maintenance decision-making, maintenance actions are also constrained by equipment states, safety regulations, risk levels, and operational conditions, so constraint-aware methods are relevant.

However, existing constraint-aware methods mostly use manually defined numerical rules, environmental feedback, or predefined safety boundaries as constraints [21,22]. There is relatively little discussion on how to extract maintenance conditions, action restrictions, and risk warnings from English technical documents and transform them into state–document–action feasibility signals [23]. This paper further focuses on how English maintenance clauses participate in candidate action constraint modeling.

2.4 Technical Document- and Knowledge-Enhanced Industrial Decision-Making Methods

Technical documents, manuals, SOPs, equipment specifications, and safety guidelines contain operating conditions, maintenance requirements, and risk warnings [24]. With advances in information retrieval, knowledge graphs, retrieval-augmented generation, and large language models, such knowledge has been used for industrial question answering, diagnosis explanation, fault analysis, and maintenance recommendation [25].

However, document-enhanced methods often treat documents as retrieval context, explanatory evidence, or generation prompts, with evaluation centered on retrieval or response quality [26,27]. Ref. [28], for example, combines power-plant documents with domain-specific instruction tuning for maintenance question answering. Its retrieved passages support answer generation rather than candidate-action feasibility. In contrast, this study assigns clause-derived labels to state–document–action triplets and uses them to supervise feasibility estimation and action ranking.

2.5 Human-in-the-Loop and Feedback Learning-Based Intelligent Manufacturing Methods

Human-in-the-loop manufacturing commonly involves operator supervision, manual correction, expert feedback, interactive learning, and model refinement [29,30]. Feedback and preference learning further convert human evaluations or risk judgments into optimization signals.

Existing methods nevertheless tend to use feedback for output correction or label refinement, with limited consideration of feedback under document constraints [31,32]. Ref. [30] uses expert labels and decision-rule feedback to improve failure prediction, but does not condition feedback optimization on technical-document feasibility. In contrast, HFLO converts preference and risk feedback into differentiable loss terms whose effects depend on the feasibility state produced by ELIC.

2.6 Research Gap and Positioning of This Paper

Existing document-enhanced and human-in-the-loop methods address complementary aspects of maintenance decision-making. Ref. [28] supports maintenance question answering through document retrieval but does not transform clauses into action-feasibility supervision. Ref. [30] introduces human feedback into failure prediction but does not jointly optimize it with document-derived action constraints. Therefore, joint supervision linking equipment states, clause-derived feasibility, and independent feedback signals remains insufficiently examined. Table 1 compares the closest methods in terms of document use, feedback, supervisory signals, and optimization objectives.

Table 1. Mechanism-Level Comparison with Closely Related Methods

Method	Document role	Feedback mechanism	Supervisory signal	Optimization and output
Ref. [28]	Retrieved QA context	Not jointly modeled	QA and instruction-tuning pairs	Retrieval-supported answer generation
Ref. [30]	Not explicitly modeled	Expert labels and decision rules	Failure labels and human feedback	Feedback-assisted failure prediction
Ours	Action-feasibility constraint	Preference and risk feedback	Task, feasibility, and controlled feedback signals	Joint optimization and action ranking

The distinction lies in the role of supervision rather than the mere use of documents or human input. Ref. [28] optimizes document-supported answer generation, whereas Ref. [30] uses human feedback to refine failure prediction. This study jointly models equipment states, clause-derived feasibility supervision, and independent feedback within a common objective and inference process. The contribution is restricted to this joint modeling under the constructed controlled protocol and does not imply general priority over existing approaches.

3. Methodology

3.1 Task Formulation

This study focuses on a maintenance-oriented decision-making task in human-centered smart manufacturing. Given the state observation of equipment at time t, denoted as $s_t \in \mathbb{R}^{d_s}$, where d_s is the dimensionality of state features, the model jointly incorporates an English technical document fragment d_t, a set of candidate maintenance actions $a_t \in \mathcal{A}$, and feedback signals h_t, to output device risk classification, RUL estimation, or an action feasibility score.

The candidate action set is defined as:

$$\mathcal{A} = \{\text{continue,monitor,warning,inspect,maintenance,shutdown}\} \quad (1)$$

These actions represent increasing intervention intensity. "Continue" indicates normal operation, "monitor" enhanced observation, "warning" abnormal-condition notification, "inspect" equipment inspection, "maintenance" corrective intervention, and "shutdown" immediate suspension.

Because the datasets do not provide maintenance-action labels, controlled rule-based labels are constructed. For MetroPT-3, $q_t \in [0,1]$ denotes the normalized fault-risk score derived from anomaly severity and failure-related intervals. For XJTU-SY, actions are mapped from the normalized RUL ratio $\rho t = \text{RULtLi}$, where L_i is the total observed lifetime of bearing i. The mappings are listed in Table 2 and remain fixed across data splits and repeated runs.

For classification, the model learns $f_\Theta(s_t, d_t, h_t) \to \hat{y}_t$, where $\hat{y}_t$denotes predicted fault risk and Θdenotes learnable parameters. For RUL prediction, it outputs $\hat{r}_t$.

Maintenance decision-making is further formulated as state–document–action feasibility estimation. For each triplet (s_t, d_t, a_t), a clause-derived label $y_t^c \in \{0,1\}$ is assigned before training. $y_t^c = 1$indicates that the action is permitted or required by the clause, whereas $y_t^c = 0$indicates that it is prohibited or conflicts with an operating condition or risk warning. This label supervises ELIC independently of its predicted feasibility score. Clauses without clear polarity are excluded from the supervised ELIC loss.

The document fragment d_t is retrieved from English manuals, SOPs, maintenance descriptions, or safety clauses and contains operating conditions, risk warnings, and action restrictions. Feedback is defined as $h_t = [h_t^p, h_t^r]$, where h_t^p and h_t^r denote preference and risk feedback. When expert feedback is unavailable, operator-like feedback is simulated using an independent risk–action protocol. Preference feedback reflects agreement with the risk-matched reference action, while risk feedback penalizes insufficiently conservative actions. The protocol does not use retrieved clauses, ELIC scores, or clause-derived labels.

The objective is to minimize prediction error while maintaining document compliance and feedback consistency. The overall loss combines task, document-constraint, and feedback-optimization terms, while performance is evaluated using task metrics, the synthetic protocol constraint violation rate (SP-CVR), and the synthetic protocol feedback alignment score (SP-FAS).

Table 2. Controlled Maintenance Action Mapping Rules

Dataset	Decision variable and interval	Action	Threshold rationale
MetroPT-3	$0 \le r_t < 0.20$	continue	Normal operating range
MetroPT-3	$0.20 \le r_t < 0.40$	monitor	Mild anomaly
MetroPT-3	$0.40 \le r_t < 0.60$	warning	Moderate anomaly
MetroPT-3	$0.60 \le r_t < 0.75$	inspect	High-risk transition
MetroPT-3	$0.75 \le r_t < 0.90$	maintenance	Severe anomaly
MetroPT-3	$0.90 \le r_t \le 1.00$	shutdown	Failure-related interval
XJTU-SY	$0.80 < \rho_t \le 1.00$	continue	Healthy stage
XJTU-SY	$0.60 < \rho_t \le 0.80$	monitor	Early degradation
XJTU-SY	$0.40 < \rho_t \le 0.60$	warning	Moderate degradation
XJTU-SY	$0.20 < \rho_t \le 0.40$	inspect	Accelerated degradation
XJTU-SY	$0.05 < \rho_t \le 0.20$	maintenance	Late degradation
XJTU-SY	$0 \le \rho_t \le 0.05$	shutdown	Critical end-of-life

For MetroPT-3, the thresholds use broader intervals below 0.60 and finer intervals above 0.60 to distinguish high-risk interventions. For XJTU-SY, the thresholds follow progressive degradation stages, with the final 5% of lifetime treated as critical. These author-defined mappings operationalize the controlled protocol rather than field-validated maintenance standards.

3.2 Overall Framework

To address the decoupling among equipment state prediction, English technical document utilization, and feedback optimization in predictive maintenance models, this study proposes an English technical document-enhanced human-in-the-loop maintenance decision-making framework.

The framework is designed based on constraint learning and feedback optimization. Its basic assumption is that maintenance conditions, risk descriptions, and action restrictions in English technical documents can provide external constraints for candidate maintenance actions, while preference and risk feedback can adjust the model's action selection behavior under complex operating conditions. The overall architecture is shown in Figure 1, where the English document-induced constraint module and the feedback-aware optimization module are detailed in Figures 2 and 3, respectively.

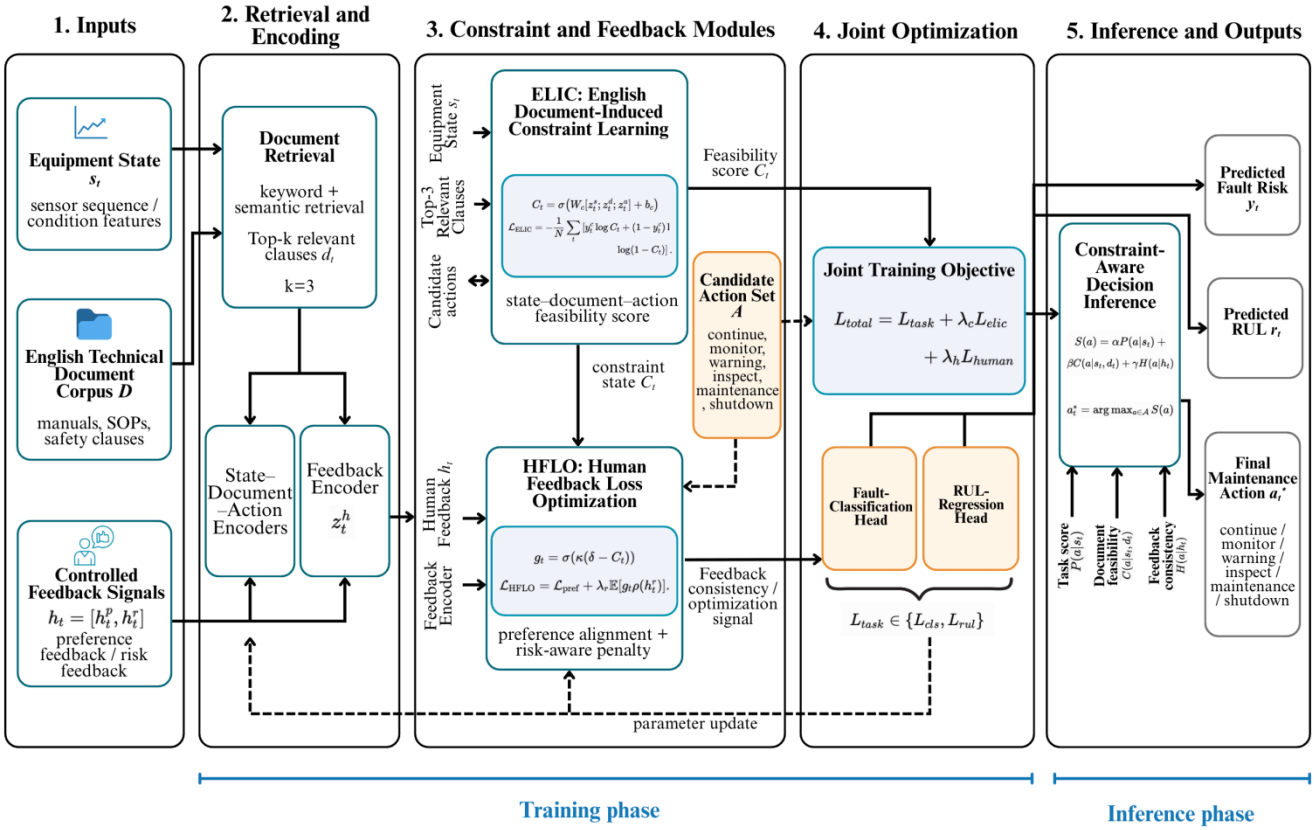

Figure 1. Overview of the English technical document-enhanced and feedback-aware maintenance decision framework

Specifically, the framework first takes the equipment state s_t as input and retrieves relevant document fragments d_t from the English technical document corpus according to the current state, fault type, or degradation stage. Then, the ELIC module jointly models s_t, d_t, and candidate actions a_t, producing a state–document–action feasibility score C_t, which is used to determine whether a candidate action satisfies the English maintenance clauses.

Different from approaches that treat documents merely as explanatory text, ELIC uses retrieved documents as constraint inputs for action feasibility evaluation, thereby allowing document content to influence action scoring and ranking.

Furthermore, the HFLO module incorporates preference feedback h_t^p and risk feedback h_t^r, and transforms them into optimization signals associated with the document constraint state. When a candidate action conflicts with English technical clauses, the constraint state C_t produced by ELIC triggers risk-based penalties. When a candidate action is consistent with preference feedback, its feedback consistency score is increased.

Finally, the model is trained using task loss, document constraint loss, and feedback optimization loss, and outputs fault risk, RUL estimation, or action feasibility results. The key aspect of the framework is that English technical clauses participate in action feasibility modeling, while feedback signals adjust the optimization direction under document-induced constraints.

3.3 English Technical-Document-Induced Constraint Learning Module

In English maintenance literature, maintenance constraints are commonly expressed through modal words such as must, should, shall, do not, warning, caution, and conditional structures such as if/when. During document filtering and encoding of d_t, the ELIC module emphasizes such rule-based semantics to support state–document–action feasibility reasoning.

The ELIC module is designed to represent maintenance conditions, action restrictions, and risk warnings in English technical documents as feasibility constraints over candidate maintenance actions. Unlike methods that treat documents only as retrieval context or explanatory evidence, ELIC uses English technical clauses as constraint inputs and introduces a state–document–action triadic modeling structure. The architecture is illustrated in Figure 2.

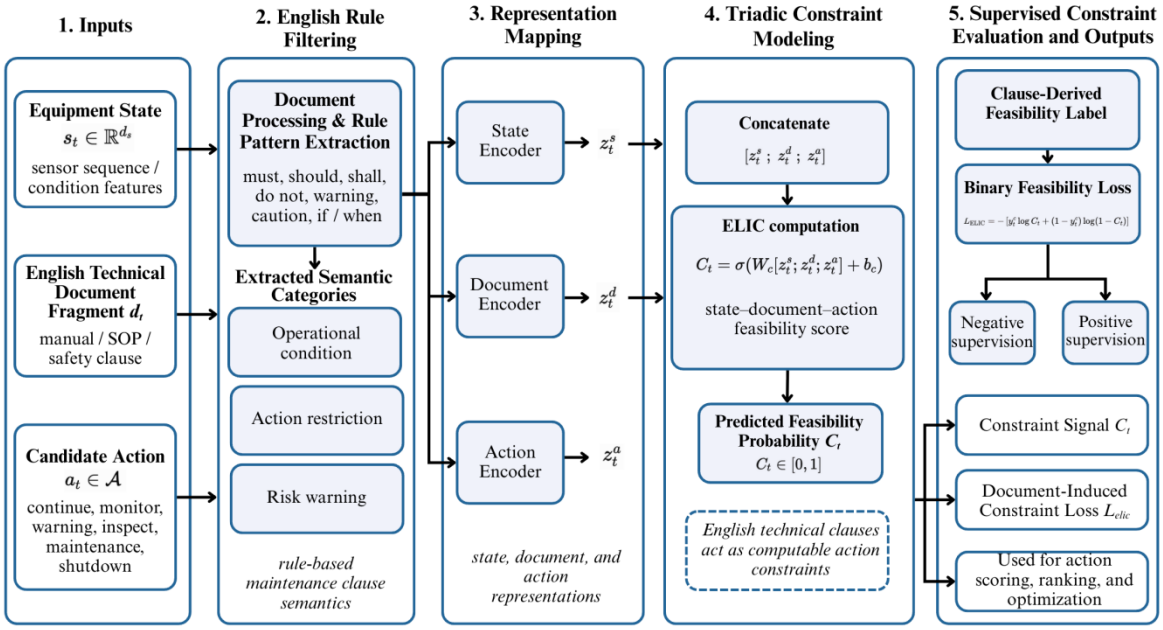

Figure 2. Architecture of the ELIC module

Given equipment state $s_t \in \mathbb{R}^{d_s}$, candidate action $a_t \in A$, and document fragment d_t, the state, action, and document are mapped into z_t^s, z_t^a, and z_t^d, respectively. Here, z_t^d encodes operating conditions, action restrictions, and risk warnings from manuals, SOPs, or safety instructions.

ELIC computes the predicted feasibility probability as:

$$C_t = \sigma\left(W_c[z_t^s; z_t^d; z_t^a] + b_c\right) \quad (2)$$

where $[\cdot;\cdot]$ denotes concatenation, W_c and b_c are learnable parameters, and $\sigma(\cdot)$ is the Sigmoid function. $C_t \in (0,1)$ estimates the feasibility of action a_t under s_t and d_t.

As defined in Section 3.1, each state–document–action triplet is assigned a clause-derived label $y_t^c \in \{0,1\}$ before training. $y_t^c = 1$ indicates that the action is permitted or required, whereas $y_t^c = 0$ indicates that it is prohibited or conflicts with an operating condition or risk warning.

The document-induced constraint loss is defined using binary cross-entropy:

$$L_{\text{ELIC}} = -\frac{1}{N}\sum_{t=1}^{N}[y_t^c \log C_t + (1-y_t^c)\log(1-C_t)]. \quad (3)$$

where $y_t^c \in \{0,1\}$ is the clause-derived feasibility label defined in Section 3.1. This polarity-aware objective increases C_t for feasible actions and decreases it for infeasible actions. ELIC is therefore supervised by clause-derived labels rather than by thresholding its own prediction. Clauses without clear polarity are excluded from the supervised loss. The threshold δ is not involved in ELIC training and is used only to convert predicted feasibility probabilities into binary decisions during inference or evaluation.

Thus, ELIC transforms English clauses into verifiable action-feasibility supervision for action scoring, ranking, and optimization.

3.4 Human Feedback Loss Optimization Module

The HFLO module models the optimization relationship between feedback signals and candidate maintenance actions under English document-induced constraint states. Unlike approaches that treat feedback as post-hoc evaluation or post-processing rules, HFLO transforms preference feedback and risk feedback into loss signals conditioned on document feasibility states, thereby enabling feedback to participate in parameter optimization. The structure is shown in Figure 3.

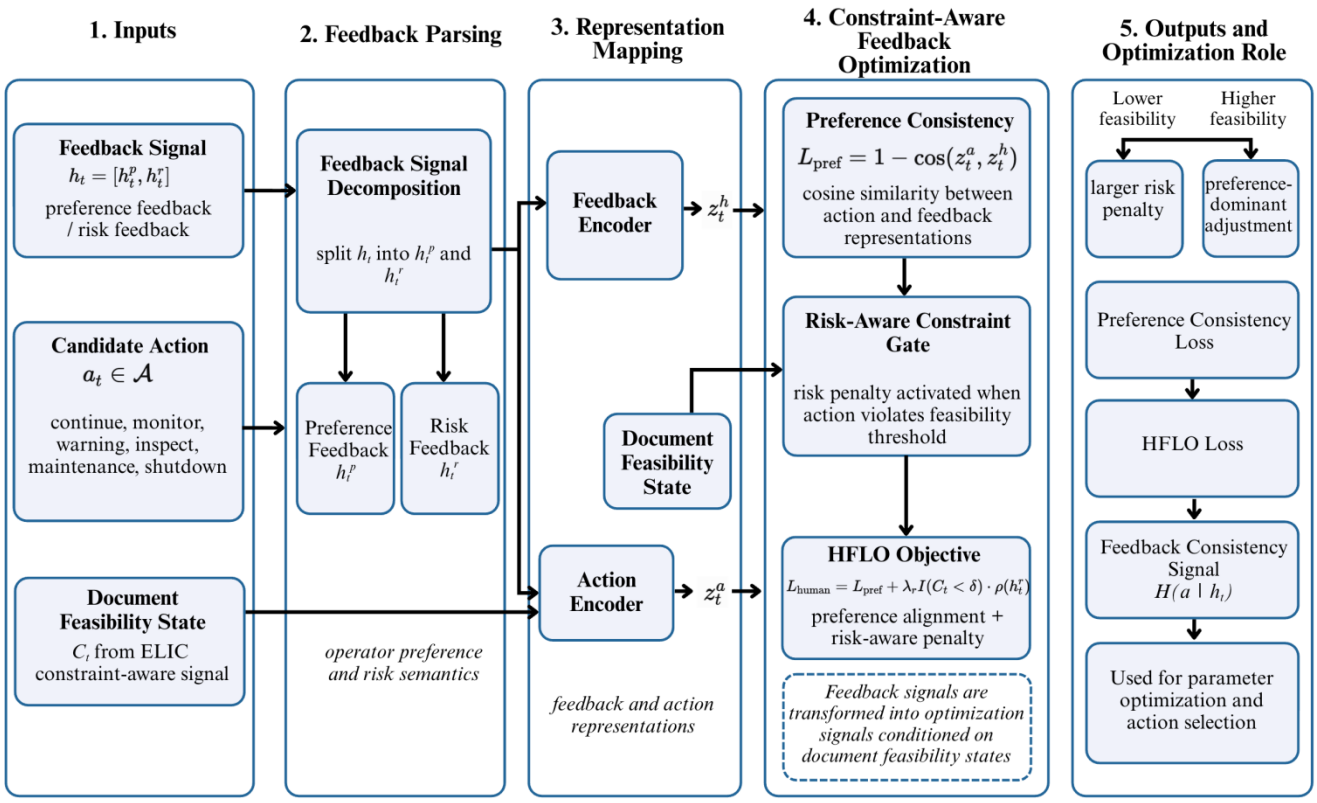

Figure 3. Architecture of the HFLO module

Given the feedback signal $h_t = [h_t^p, h_t^r]$, where h_t^p denotes preference feedback and h_t^r denotes risk feedback, the feedback signal is mapped into a semantic representation z_t^h, while candidate actions are mapped into z_t^a. To avoid coupling feedback supervision with document-induced constraints, h_t^p and h_t^r are generated independently of the English clause corpus. Neither feedback component uses $C(a \mid s_t, d_t)$, retrieved clauses, or clause-derived feasibility labels during construction. The preference consistency loss is defined as:

$$L_{pref} = 1 - \cos(z_t^a, z_t^h) \quad (4)$$

where $\cos(\cdot)$ denotes cosine similarity, used to measure the consistency between action representations and preference feedback representations.

Furthermore, HFLO incorporates the document feasibility state C_t produced by ELIC, with risk-feedback strength adjusted continuously according to predicted action feasibility. A differentiable smooth feasibility gate is defined as $\mathbf{g_t} = \boldsymbol{\sigma}(\boldsymbol{\kappa}(\boldsymbol{\delta} - \mathbf{C_t}))$, where $\sigma(\cdot)$ is the Sigmoid function and $\kappa > 0$ controls the transition slope. The feedback optimization loss is then defined as:

$$\mathbf{L_{human}} = \mathbf{L_{pref}} + \boldsymbol{\lambda_r}\mathbb{E}[\mathbf{g_t} \cdot \boldsymbol{\rho}(\mathbf{h_t^r})]. \quad (5)$$

where λ_r is the risk-feedback weighting coefficient, $g_t \in (0,1)$ is the smooth feasibility gate, and $\rho(h_t^r)$ denotes the penalty derived from risk feedback. When C_t decreases below δ, g_t increases smoothly and assigns greater weight to the risk-feedback penalty. When C_t is high, the penalty is reduced, and feedback mainly influences action selection through the preference-consistency term. Unlike a hard indicator function, the smooth gate is differentiable with respect to C_t, allowing continuous gradient propagation through ELIC and HFLO. Thus, HFLO links feedback optimization with English document feasibility states. Removing this module would weaken the model's ability to represent the constructed preference and risk signals.

3.5 Joint Maintenance Decision Objective

After completing document constraint modeling in ELIC and feedback optimization in HFLO, this study constructs a joint maintenance decision objective, enabling task prediction, English document constraints, and feedback optimization to jointly contribute during training. This objective does not treat documents and feedback as post-processing rules, but incorporates them into parameter updates, thereby influencing candidate action feasibility evaluation.

For the fault classification task, the task loss is defined as:

$$L_{cls} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \quad (6)$$

where N is the number of samples, y_i denotes the ground-truth fault label, and $\hat{y}_i$ is the predicted fault probability.

For the RUL prediction task, the loss is defined as:

$$L_{rul} = \frac{1}{N}\sum_{i=1}^{N}(r_i - \hat{r}_i)^2 \quad (7)$$

where r_i and $\hat{r}_i$ denote the ground-truth and predicted remaining useful life, respectively. Depending on the task type, the base task loss is denoted as $L_{task} \in \{L_{cls}, L_{rul}\}$.

On this basis, the document constraint loss from ELIC and the feedback loss from HFLO are incorporated into the total objective:

$$L_{total} = L_{task} + \lambda_c L_{elic} + \lambda_h L_{human} \quad (8)$$

where λ_c and λ_h are weighting coefficients for the document constraint loss and feedback loss, respectively. This objective enables English technical constraints and feedback consistency requirements to be incorporated into maintenance decision modeling during training while optimizing predictive performance.

3.6 Constraint-Aware Decision Inference

During the inference stage, this study designs a constraint-aware maintenance action selection mechanism to transform model outputs into final maintenance decisions. Candidate maintenance actions are required not only to achieve high task prediction scores, but also to satisfy English document constraints and remain consistent with feedback signals. Therefore, the final action selection does not rely on a single prediction result, but integrates task scores, document feasibility, and feedback consistency.

Given the candidate action set $\mathcal{A}$, the comprehensive decision score for each action $a \in \mathcal{A}$ is computed as:

$$S(a) = \alpha P(a \mid s_t) + \beta C(a \mid s_t, d_t) + \gamma H(a \mid h_t) \quad (9)$$

where $P(a \mid s_t)$ denotes the task-based action score. For MetroPT-3, the predicted fault-risk score $\hat{q}_t \in [0,1]$ is mapped to the six action intervals defined in Table 2. For XJTU-SY, the predicted RUL $\hat{r}_t$ is first normalized as $\hat{\rho}_t = \hat{r}_t / L_i$, where L_i is the total observed lifetime of bearing i, and then mapped to the corresponding degradation interval in Table 2. Accordingly, $P(a \mid s_t)$ is assigned according to whether the predicted risk or normalized RUL falls within the fixed interval associated with action a. The same thresholds are used across training, validation, testing, and repeated runs.

$C(a \mid s_t, d_t)$ represents the document feasibility score produced by ELIC, and $H(a \mid h_t)$ represents the feedback consistency score produced by HFLO, which can be formulated as:

$$H(a \mid h_t) = \cos(F_a(a), G_h(h_t)) \quad (10)$$

where $F_a(a)$ is the action encoder and $G_h(h_t)$ is the feedback encoder. α, β, and γ are weighting coefficients, which can be selected on a validation set and satisfy $\alpha + \beta + \gamma = 1$. The final maintenance action is determined by:

$$a_t^* = \arg\max_{a \in \mathcal{A}} S(a) \quad (11)$$

This mechanism allows document constraints and feedback signals to influence final action ranking. An action with a high task score receives a lower overall score when it conflicts with an English technical clause or risk feedback. Thus, the final action is selected using the fixed task-to-action mappings in Table 2 together with document feasibility and feedback consistency, ensuring that the controlled action protocol is applied consistently during inference.

3.7 Empirical Optimization Stability Analysis

To evaluate the stability of the joint optimization process without imposing unverified boundedness assumptions, this study analyzes the observed training dynamics over five independent runs. The total loss, synthetic protocol constraint violation rate (SP-CVR), and synthetic protocol feedback alignment score (SP-FAS) are recorded at each training epoch.

For metric m at epoch e, its run-wise mean is calculated as:

$$\bar{m}_e = \frac{1}{R} \sum_{r=1}^{R} m_e^{(r)} \quad (12)$$

where $m_e^{(r)}$ denotes the value obtained in run r, and $R = 5$. The corresponding run-wise variance is:

$$V_e(m) = \frac{1}{R-1} \sum_{r=1}^{R} \left(m_e^{(r)} - \bar{m}_e\right)^2 \quad (13)$$

The standard deviation used for the uncertainty bands in the training curves is calculated as:

$$SD_e(m) = \sqrt{V_e(m)} \quad (14)$$

The loss weights λ_c and λ_h are selected on the validation set and remain fixed across repeated runs.

This analysis does not claim global convergence or theoretically bounded gradient variance. Instead, it characterizes optimization behavior using the observed mean and variance of the training curves.

3.8 Training and Inference Procedure

This section summarizes the overall training and constraint-aware inference procedure of the proposed English technical document-enhanced human-in-the-loop maintenance decision-making framework. The overall process includes equipment state encoding, English technical document retrieval, document-induced constraint learning, feedback-aware optimization, joint parameter updating, and final maintenance action inference. Algorithm 1 presents the complete procedure.

Algorithm 1. Training and inference procedure of the proposed framework

Input: training samples $\{(s_t, a_t, y_t, r_t, y_t^c)\}_{t=1}^{N}$, document library D, action set A, feedback h_t, and weights λ_c, λ_h
Output: optimized parameters Θ, predictions $\hat{y}_t, \hat{r}_t$, feasibility score C_t, and action a_t^*
1: Initialize Θ.
2: For each training epoch do
3: For each sample t do
4: Retrieve d_t from D and obtain its clause-derived label

y_t^c.
5: Encode s_t, d_t, and $a_t$as z_t^s, z_t^d, and z_t^a.
6: Compute feasibility probability $C_t$using ELIC.
7: Compute

$$L_{\text{ELIC}} = -[y_t^c \log C_t + (1 - y_t^c)\log(1 - C_t)].$$

8: Compute feedback loss L_{human} using HFLO.
9: Compute task loss L_{task}.
10: Update Θ by minimizing

$$L_{\text{total}} = L_{\text{task}} + \lambda_c L_{\text{ELIC}} + \lambda_h L_{\text{human}}.$$

11: End for
12: End for Inference Stage
13: Compute $S(a)$for each $a \in A$.
14: Select

$$a_t^* = \arg\max a \in \text{A } S(a).$$

15: Return $\hat{y}_t, \hat{r}_t, C_t$,and a_t^*.

This algorithm indicates that English technical documents and feedback signals are not only used as post-hoc explanations, but also participate in parameter updates during training and influence final maintenance action ranking during inference. This is consistent with the proposed document-constrained and feedback-aware maintenance decision-making mechanism.

4. Experiments and Results

4.1 Experimental Settings

Datasets and Preprocessing

This study uses the MetroPT-3 and XJTU-SY Bearing datasets. MetroPT-3 contains 1,516,948 samples with 15 features collected from a metro-train air-compressor system between February and August 2020 and is used for fault-risk classification. XJTU-SY contains run-to-failure data from 15 rolling bearings and is used for remaining useful life (RUL) prediction.

Preprocessing includes time ordering, missing-value checking, outlier clipping, Z-score normalization, and sliding-window construction. Outliers are clipped at the 1st and 99th percentiles of the training set, whose statistics are also used for normalization. MetroPT-3 retains all 15 features and uses 128-step windows with a stride of 64. Samples are chronologically divided into training, validation, and test sets at a ratio of 70:10:20.

For XJTU-SY, each horizontal and vertical vibration signal is divided into 2,048-point segments. Six statistical features, mean, standard deviation, root mean square, peak-to-peak value, skewness, and kurtosis, are extracted from each channel, producing 12 features per segment. Sequences of 32 segments with a stride of 8 are constructed. Bearings 1–3 under each operating condition are used for training, Bearing 4 for validation, and Bearing 5 for testing, yielding 9, 3, and 3 bearings, respectively. This bearing-level split prevents data leakage.

Because neither dataset provides maintenance-action labels, English clauses, or expert feedback, a controlled maintenance protocol is constructed. For MetroPT-3, the normalized fault-risk score q_t is derived from anomaly severity and failure-related intervals, and actions are assigned using Table 2. For XJTU-SY, actions are assigned from the normalized RUL ratio ρ_t and the degradation intervals in Table 2. The thresholds remain fixed across data splits and repeated runs. These labels support controlled evaluation rather than field-observed maintenance decisions and provide a common task-to-action mapping for all methods.

An English technical clause library is constructed from public manuals, SOPs, safety instructions, and fault-handling rules. It contains 360 polarity-explicit clauses, including 120 operating-condition clauses, 120 action-limitation clauses, and 120 risk-warning clauses. For each state–document–action triplet (s_t, d_t, a_t), a fixed clause-derived feasibility label $y_t^c \in \{0,1\}$ is assigned before training. Permitted or required actions are labeled $y_t^c = 1$, whereas prohibited or conflicting actions are labeled $y_t^c = 0$. Clauses without clear polarity are excluded. These labels are independent of $C_t$and remain fixed across splits and runs. The library is used for retrieval, label construction, and ELIC training, but not for feedback simulation. During evaluation, the fixed labels are accessed only by an external evaluator.

To reduce circular validation between ELIC and HFLO, feedback is generated using an independent risk–action protocol that does not access retrieved clauses, ELIC outputs, or clause-derived labels. The six actions are assigned ordinal levels from 0 to 5 in the order continue, monitor, warning, inspect, maintenance, and shutdown. The reference action a_t^* is obtained from the fixed task-to-action mapping in Table 2. Table 3 then defines preference and risk feedback according to the ordinal relationship between the candidate action and a_t^*.

Table 3. Independent Feedback Simulation Rules

Feedback type	Rule	Signal
Preference	Candidate action equals a_t^{ref}	Positive $(h_t^p = 1)$
Preference	Difference of one level	Neutral $(h_t^p = 0)$
Preference	Difference of at least two levels	Negative $(h_t^p = -1)$
Risk	Reference action is inspect, maintenance, or shutdown, and candidate action is less conservative	Penalty $(h_t^r = 1)$
Risk	Candidate action equals or exceeds the reference level	No penalty $(h_t^r = 0)$
Risk	Reference action is continue, monitor, or warning	No penalty $(h_t^r = 0)$

Baseline Methods

This study compares eight baselines with the proposed method, covering traditional machine learning, temporal models, transformer-based prediction, document-enhanced maintenance, and human-in-the-loop methods.

All baselines are adapted to MetroPT-3 fault-risk classification and XJTU-SY RUL regression while retaining their original modeling mechanisms. Random Forest and XGBoost use classification or regression objectives according to the dataset. Sequence-based models retain their original temporal encoders and use a binary classification head for MetroPT-3 and a scalar regression head for XJTU-SY. All methods use identical data splits, preprocessing, state features, input windows, action mappings, validation settings, random seeds, and external evaluators. Tree-based methods receive flattened state windows, whereas neural models receive the original sequential windows.

The RAG-based maintenance baseline adapted from Ref. [28] uses the shared TCN–Transformer backbone, the corresponding dataset-specific task head, and retrieved clauses, but does not access clause-derived labels, ELIC losses, or feedback. The human-in-the-loop PdM baseline adapted from Ref. [30] uses the same backbone and task heads together with the Table 3 feedback signals but does not access English clauses. Thus, these two controlled baselines differ from the proposed method mainly in information access and learning objectives rather than backbone capacity.

Task-only methods map their predictions to actions using Table 2. The model-agnostic evaluator applies fixed clause labels and Table 3 feedback rules without accessing internal scores or embeddings. Table 4 summarizes task adaptation, information access, and model capacity. Paired parameter counts correspond to MetroPT-3/XJTU-SY. The shared frozen all-MiniLM-L6-v2 encoder has 22.7 M parameters and is excluded from trainable counts.

Table 4. Baseline Adaptation, Information Access, and Model Capacity

Method	MetroPT-3	XJTU-SY	State	Clauses	Clause labels	Feedback	Capacity
Random Forest [33]	Classification	Regression	Yes	No	No	No	500 trees; depth 20
XGBoost [34]	Classification	Regression	Yes	No	No	No	500 trees; depth 8
LSTM [35]	Classification	Regression	Yes	No	No	No	2 layers, 128 hidden; 0.23/0.22 M
Transformer-RUL [36]	Classification	Regression	Yes	No	No	No	3 layers, d=128, 4 heads; 0.41/0.40 M
Hierarchical Transformer [37]	Classification	Regression	Yes	No	No	No	4 layers, d=128, 4 heads; 0.60/0.59 M
TCN–Transformer [38]	Classification	Regression	Yes	No	No	No	3 TCN blocks + 3 Transformer layers; 0.96/0.95 M
RAG-style maintenance [28]	Classification	Regression	Yes	Yes	No	No	1.12/1.11 M
Human-in-the-loop PdM [30]	Classification	Regression	Yes	No	No	Yes	1.01/1.00 M
Ours	Classification	Regression	Yes	Yes	Training only	Yes	1.15/1.14 M

Implementation Details

The model is implemented in PyTorch and optimized using Adam with an initial learning rate of $1 \times 10^{-3}$and a batch size of 64. Experiments use an Intel Core i7-12700K CPU, 32 GB RAM, and an NVIDIA RTX 3060 GPU. Neural models are trained for 50 epochs with a weight decay of 1×10^{-4} and dropout of 0.1.

The shared backbone contains three 64-channel residual TCN blocks and three Transformer layers with a representation dimension of 192, four attention heads, and a feed-forward dimension of 384. It is used by TCN–Transformer, the RAG-based model, the human-in-the-loop baseline, and the proposed method.

English clauses are retrieved through keyword and semantic matching, with the top $k = 3$ clauses selected per sample. Document-enabled methods use the same frozen all-MiniLM-L6-v2 encoder to produce 384-dimensional embeddings. Keyword and semantic scores are equally weighted, and clause embeddings are projected to 192 dimensions before fusion.

The loss weights are $\lambda_c = 0.4$, $\lambda_h = 0.3$, and $\lambda_r = 0.5$, with $\kappa = 10$. During inference, task, document-feasibility,

and feedback-consistency scores are weighted by 0.5, 0.3, and 0.2, respectively, and the feasibility threshold is 0.5. All values are selected on the validation sets and fixed across runs.

ELIC accesses equipment states, candidate actions, and retrieved clauses. The feedback simulator accesses only the risk or normalized-RUL stage, candidate-action level, and Table 2 reference action. The external evaluator accesses only the final action, fixed clause label, and Table 3 feedback signals, not C_i or internal representations. It is unavailable during training and inference.

All baselines, controlled variants, and ablations are run using the same five random seeds: 13, 21, 42, 87, and 102. Data splits, preprocessing, input windows, clause corpus, feedback rules, action thresholds, initialization protocol, and the external evaluator are fixed across methods. All tabulated and plotted statistics are computed from these five runs and reported as mean ± standard deviation.

Evaluation Metrics

Three categories of metrics are used. F1 is used for MetroPT-3, and RMSE is used for XJTU-SY. Because neither dataset contains field-observed maintenance actions or expert evaluations, the remaining metrics are reported strictly as synthetic protocol consistency measures.

For each test sample, the evaluated method produces a final action a_i^*. Task-only baselines obtain this action using the fixed intervals in Table 2. A model-agnostic external evaluator then compares a_i^* with the fixed clause labels and feedback rules. Because the evaluator receives only the final action and fixed annotations, it cannot favor methods that internally use documents or feedback.

The synthetic protocol constraint violation rate is:

$$\text{SP-CVR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(y^c_{i,a_i^*} = 0\right), \tag{15}$$

where y^c_{i,a_i^*} is the fixed feasibility label of the selected action. A lower SP-CVR indicates fewer violations under the constructed clause protocol. It does not use the model-generated score C_i.

The synthetic protocol feedback alignment score is:

$$\text{SP-FAS} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{h_i^p(a_i^*)+1}{4} + \frac{1-h_i^r(a_i^*)}{2}\right], \tag{16}$$

where $h_i^p(a_i^*) \in \{-1,0,1\}$ and $h_i^r(a_i^*) \in \{0,1\}$ are generated using Table 3. A higher SP-FAS indicates greater consistency with the constructed preference and risk protocol. It does not use action or feedback embeddings.

SP-CVR and SP-FAS are computed identically for all methods and measure consistency with the controlled synthetic protocol rather than actual operator judgments or field maintenance quality.

4.2 Main Experimental Results

Table 5 and Table 6 report the main results on the MetroPT-3 and XJTU-SY datasets, respectively. All results are reported as mean ± standard deviation over five runs. For MetroPT-3, F1, SP-CVR, and SP-FAS evaluate classification performance, synthetic protocol constraint violations, and synthetic protocol feedback consistency, respectively. For XJTU-SY, RMSE, SP-CVR, and SP-FAS evaluate RUL error and consistency with the constructed constraint and feedback protocols.

Table 5. Main results on MetroPT-3

Method	F1 ↑	SP-CVR ↓	SP-FAS ↑
Random Forest	0.842±0.006	0.214±0.011	0.681±0.014
XGBoost	0.876±0.005	0.186±0.010	0.704±0.012
LSTM	0.861±0.007	0.201±0.012	0.697±0.015
Transformer-based RUL model	0.872±0.006	0.193±0.011	0.716±0.013
Hierarchical Transformer	0.884±0.005	0.179±0.009	0.731±0.012
TCN–Transformer	0.887±0.004	0.174±0.010	0.738±0.011
RAG-based maintenance	0.879±0.006	0.151±0.008	0.762±0.010
Human-in-the-loop PdM	0.874±0.006	0.168±0.009	0.806±0.009
Ours	0.895±0.004	0.124±0.007	0.798±0.008

Table 6. Main results on XJTU-SY

Method	RMSE ↓	SP-CVR ↓	SP-FAS ↑
Random Forest	15.82±0.31	0.226±0.012	0.662±0.016
XGBoost	14.36±0.26	0.204±0.010	0.681±0.014
LSTM	13.91±0.24	0.213±0.011	0.689±0.013
Transformer-based RUL model	13.42±0.22	0.196±0.010	0.701±0.012
Hierarchical Transformer	12.91±0.19	0.181±0.009	0.717±0.012
TCN–Transformer	12.56±0.18	0.176±0.009	0.724±0.011
RAG-based maintenance model	13.08±0.21	0.149±0.008	0.751±0.010
Human-in-the-loop maintenance model	13.24±0.23	0.163±0.009	0.792±0.009
Ours	12.68±0.17	0.121±0.007	0.785±0.008

From Table 5, the proposed method achieves the highest F1 and lowest SP-CVR on MetroPT-3, whereas the human-in-the-loop baseline achieves the highest SP-FAS. The F1 improvement over TCN–Transformer is small, while the SP-CVR reduction relative to the RAG-based model suggests that document retrieval alone does not ensure consistency with the constructed clause protocol.

On XJTU-SY, TCN–Transformer achieves the lowest RMSE of 12.56±0.18, and the human-in-the-loop baseline achieves the highest SP-FAS of 0.792±0.009. The proposed method obtains an RMSE of 12.68±0.17, an SP-FAS of 0.785±0.008, and the lowest SP-CVR of 0.121±0.007. Its main advantage therefore lies in reducing synthetic protocol violations rather than achieving the best result on every metric. These results do not represent agreement with field-observed decisions or actual operator judgments.

Figure 4 shows the trade-off between task performance and SP-CVR on both datasets. The proposed method lies near a favorable trade-off region, although it is not best on every metric, consistent with protocol-aware action selection rather than pure error minimization.

Figure 5 reports the epoch-wise means and ±1 standard-deviation bands of task loss, validation SP-CVR, and validation SP-FAS over five runs. The curves are used to assess empirical stabilization and run-to-run variation, not theoretical convergence or bounded gradient variance.

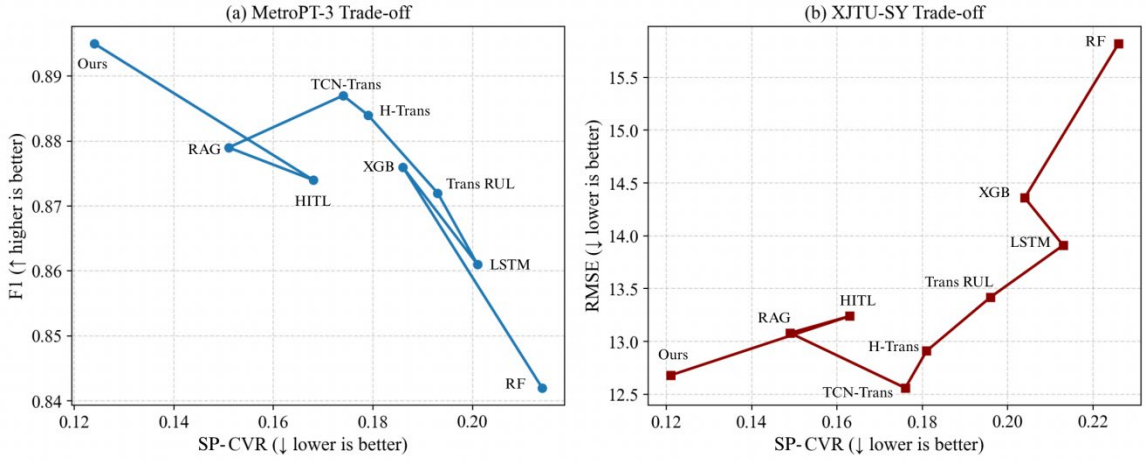

Figure 4. Trade-off analysis of task performance and constraint violation across different methods
(a) Trade-off between F1 and SP-CVR on the MetroPT-3 dataset
(b) Trade-off between RMSE and SP-CVR on the XJTU-SY dataset

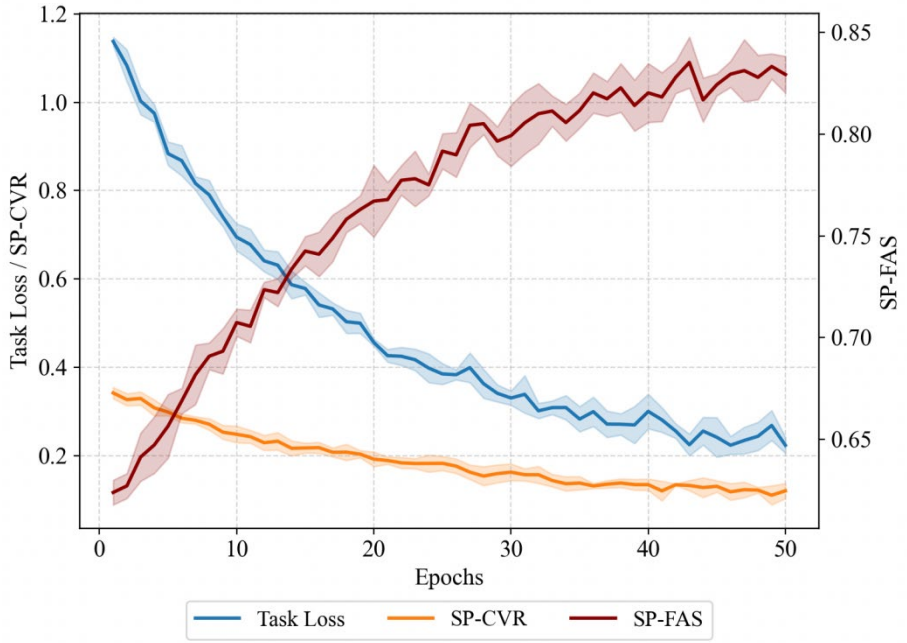

Figure 5. Empirical Training Dynamics over Five Independent Runs; Solid Curves Denote Epoch-Wise Means, and Shaded Bands Denote ±1 Standard Deviation

4.3 Statistical Significance Analysis

To evaluate statistical reliability, XGBoost, TCN–Transformer, and the RAG-based maintenance model are compared with the proposed method using paired t-tests over the same five runs, with $\alpha = 0.05$. Table 7 reports $\Delta =$ Ours − Baseline, p-values, and 95% confidence intervals. Positive F1 differences favor the proposed method, whereas negative RMSE and SP-CVR differences indicate lower error or fewer violations.

Table 7 shows significant F1 improvements over XGBoost and TCN–Transformer on MetroPT-3 and significant SP-

CVR reductions relative to the RAG-based model on both datasets. On XJTU-SY, the RMSE reduction relative to XGBoost is significant. Compared with TCN–Transformer, however, the proposed method has an RMSE that is 0.12 higher, and the difference is not statistically significant ($p = 0.087$, 95% CI: $[-0.02, 0.26]$). Thus, this comparison provides no evidence of improved RUL prediction over TCN–Transformer.

Table 7. Paired t-Tests and 95% Confidence Intervals over Five Unified Runs

Dataset	Comparison	Metric	(\Delta)	(p)-value	95% CI
MetroPT-3	Ours vs XGBoost	F1 ↑	+0.019	0.004	[0.011, 0.027]
MetroPT-3	Ours vs TCN–Transformer	F1 ↑	+0.008	0.031	[0.002, 0.014]
MetroPT-3	Ours vs RAG-based maintenance model	SP-CVR ↓	-0.027	0.006	[-0.039, -0.015]
XJTU-SY	Ours vs XGBoost	RMSE ↓	-1.68	0.003	[-2.21, -1.15]
XJTU-SY	Ours vs TCN–Transformer	RMSE ↓	+0.12	0.087	[-0.02, 0.26]
XJTU-SY	Ours vs RAG-based maintenance model	SP-CVR ↓	-0.028	0.005	[-0.041, -0.016]

These observations are consistent with Tables 5 and 6, suggesting that the main advantage of the proposed method lies in reducing violations under the synthetic clause protocol while maintaining performance close to strong temporal models. Overall, the statistical results support the contribution of English technical documents to synthetic protocol constraint consistency, rather than superiority across all metrics. Figure 6 presents the 95% confidence intervals to illustrate the direction and variability of the differences.

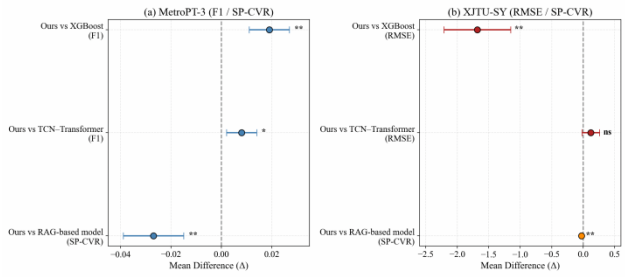

Figure 6. Paired mean differences and 95% confidence intervals over five unified runs. Differences are calculated as Ours minus Baseline, and intervals crossing zero indicate non-significant differences: (a) F1; (b) RMSE and SP-CVR

4.4 Representation and Feature Space Analysis

To analyze the impact of ELIC and HFLO on the representation space, this study conducts visualization and quantitative evaluation of joint state–document–action representations. This experiment examines whether the model forms a clearer clause-derived feasibility boundary in the latent space rather than comparing final predictive performance (Table 8).

Three settings are considered: Baseline, w/o ELIC, and Ours. Baseline uses only state features, while w/o ELIC removes English document constraints during training but retains feedback optimization.

Table 8 reports the representation metrics as mean ± standard deviation over the same five runs used in the main experiments. The proposed method shows higher average Silhouette Scores and centroid distances and lower average intra-class variance on both datasets. These metrics describe representation-level separability only and do not establish superior predictive performance or field-level decision quality.

Table 8. Representation Separability Metrics over Five Unified Runs

Dataset	Method	Silhouette Score ↑	Centroid Distance ↑	Intra-class Variance ↓
MetroPT-3	Baseline	0.214 ± 0.013	1.38 ± 0.08	0.742 ± 0.019
MetroPT-3	w/o ELIC	0.247 ± 0.011	1.56 ± 0.10	0.711 ± 0.017
MetroPT-3	Ours	0.304 ± 0.009	1.78 ± 0.07	0.628 ± 0.015
XJTU-SY	Baseline	0.198 ± 0.014	1.26 ± 0.09	0.781 ± 0.022
XJTU-SY	w/o ELIC	0.236 ± 0.012	1.39 ± 0.08	0.735 ± 0.018
XJTU-SY	Ours	0.281 ± 0.010	1.68 ± 0.11	0.659 ± 0.016

Figure 7 presents a nine-panel t-SNE visualization of action feasibility representations under Baseline, w/o ELIC, and Ours. In the Baseline setting, different action states exhibit substantial overlap. After incorporating feedback optimization (w/o ELIC), the representation space improves to some extent, although feasible and infeasible actions remain partially mixed. In the proposed method, the distribution becomes more compact, and action category overlap is further reduced. Since t-SNE is a dimensionality reduction visualization technique, it is only used as auxiliary evidence rather than a standalone performance indicator.

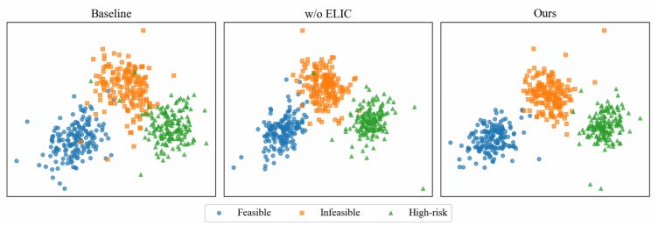

Figure 7. Representation space of maintenance actions

Figure 8 illustrates a matrix heatmap of embedding similarity between states, English documents, and candidate actions. The horizontal axis represents candidate actions, while the vertical axis corresponds to document clause types, including operating condition, action limitation, and risk warning. The results show that risk warning is more strongly associated with maintenance and shutdown, while operating condition is more related to continue and monitor. Unlike the clause-based evidence scores in Section 4.5, Figure 8 primarily reflects semantic alignment at the representation level.

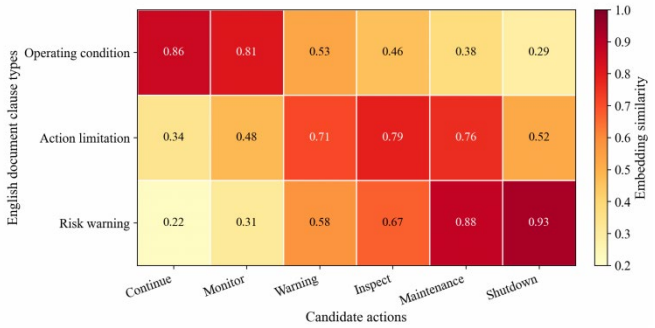

Figure 8. Embedding similarity between English Clause Types and Candidate Actions

4.5 Interpretability and Constraint Evidence Analysis

To analyze the interpretability of model decisions, this study investigates the evidence sources of recommended actions from English technical document constraints and the simulated feedback protocol. This section does not directly evaluate predictive performance, but examines whether the outputs form a traceable "clause–feedback–action" evidence chain.

Table 9 presents representative cases. The action adjustments of the proposed method can generally be associated with specific English clause types, whereas the baseline mainly relies on sensor-based prediction and cannot explicitly attribute decisions to document constraints. These cases illustrate evidence-tracing patterns under the controlled protocol rather than field-validated decision explanations.

Table 9. Constraint evidence case table

Case	English Clause Type	Baseline Action	Ours Action	Evidence Explanation
1	risk warning	monitor	shutdown	High-risk clauses increase weighting toward shutdown actions
2	action limitation	inspect	maintenance	Action restriction clauses adjust maintenance level
3	operating condition	continue	monitor	Operating condition clauses suppress continued operation
4	mixed clause	warning	inspect	A balanced action is derived under multiple clause types

Figure 9 presents a relevance heatmap between English clauses and candidate actions, showing how clause types affect action scores. Unlike the embedding similarity in Figure 8, Figure 9 focuses on score-level evidence used in action ranking. Risk warnings show stronger relevance to maintenance and shutdown, whereas operating conditions are more closely related to continue and monitor. These patterns indicate associations within the learned scoring mechanism rather than verified causal effects of individual clauses.

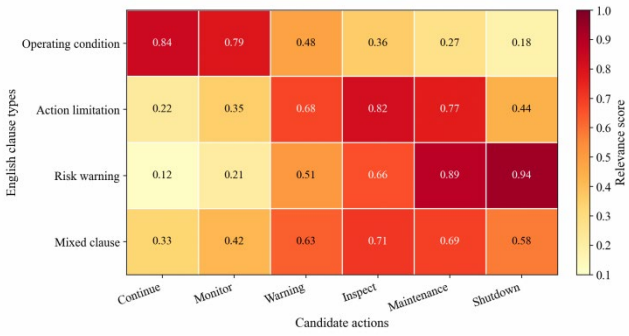

Figure 9. Relevance heatmap between English Clauses and Candidate Actions

Figure 10 compares SP-FAS distributions for the full model, w/o HFLO, w/o risk feedback, and w/o preference feedback. Removing HFLO reduces SP-FAS and increases run-to-run variation. Removing either feedback component also lowers consistency with the simulated feedback protocol, indicating that both signals contribute to action selection under the controlled evaluation setting.

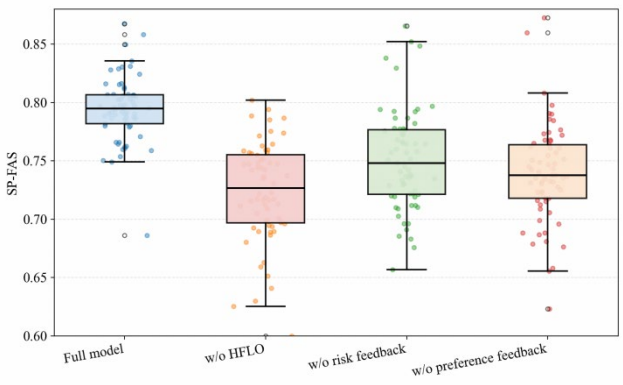

Figure 10. SP-FAS distributions under selected ablation settings

Figure 11 presents a nine-panel visualization of decision evidence, illustrating the process from sensor input and English clause retrieval to final action ranking. Panels (a)–(b) show input-state variations, (c) shows retrieved-clause relevance, (d)–(e) show baseline and proposed action scores,(f) shows the document feasibility probability C_t, (g) shows the model-internal feedback-consistency term $H(a \mid h_t)$, and (h)–(i) show final ranking and evidence-contribution decomposition.

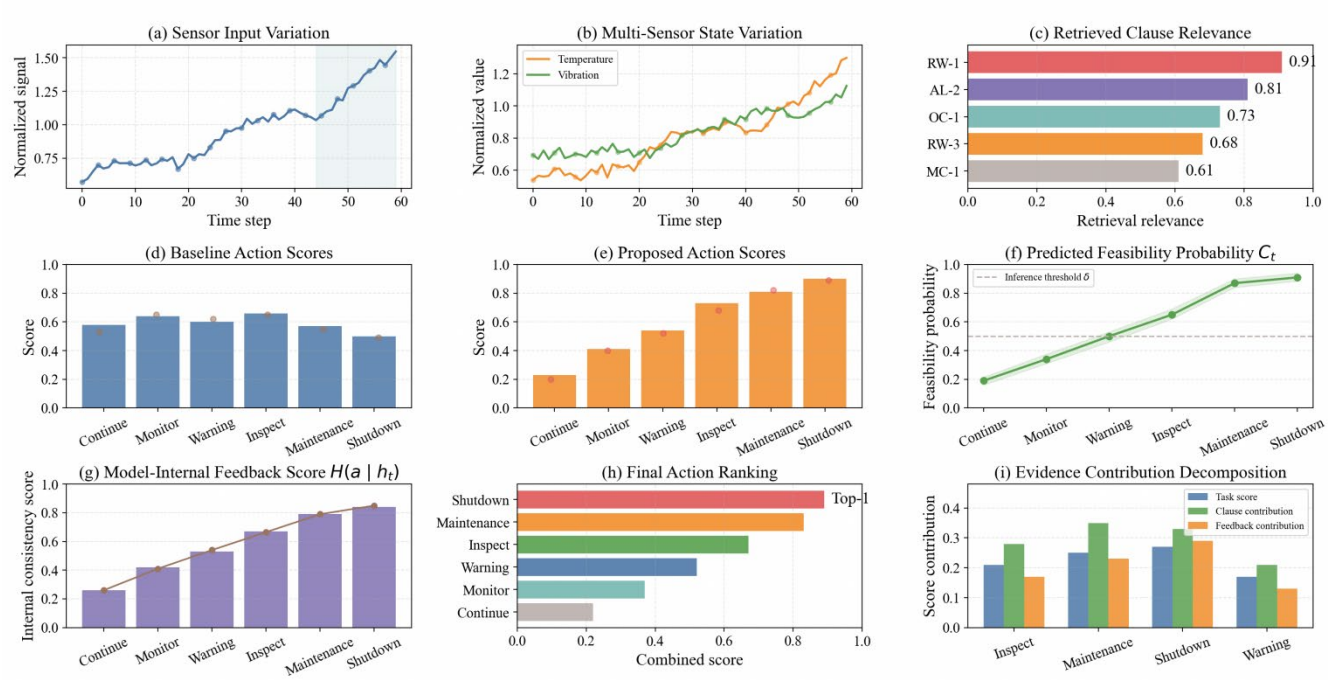

Figure 11. Nine-panel visualization of decision evidence
(a) Sensor input variation; (b) multi-sensor state variation; (c) retrieved clause relevance; (d) baseline action scores; (e) proposed action scores; (f) document feasibility C_t; (g) feedback consistency $H(a \mid h_t)$; (h) final action ranking; (i) evidence contribution decomposition

Overall, ELIC and HFLO provide traceable clause-based and feedback-based evidence for maintenance-action ranking. However, the evidence reflects the constructed clause library, simulated feedback protocol, and learned scoring mechanism and should not be interpreted as expert-validated explanations or field-observed decision rationales.

4.6 Error and Failure Case Analysis

To analyze the applicability boundary of the proposed method, this section investigates failure modes of ELIC and HFLO from two perspectives: failure-type summarization and representative case visualization.

This experiment is not intended to demonstrate performance improvements, but to illustrate potential errors under English clause retrievalambiguity, feedback conflicts, and boundary degradation states.

Table 10 shows that the main failure sources include English clause retrieval deviation, semantic ambiguity, feedback conflicts, and boundary RUL samples. Retrieval errors affect constraint construction in ELIC, leading to inappropriate feasibility scores for candidate actions. Feedback conflicts weaken the regulatory effect of HFLO on action ranking, resulting in reduced consistency with the synthetic feedback protocol.

Table 10. Failure case categories and causes

Failure Type	Cause	Affected Metric	Observation
Retrieval error	Inaccurate English clause retrieval	SP-CVR / SP-FAS	Misaligned constraint signals
Ambiguous clause	Semantic ambiguity in technical clauses	SP-CVR	Unstable action constraints
Feedback conflict	Conflict between preference and risk signals	SP-FAS	Decreased protocol consistency
Boundary RUL case	Critical degradation state	RMSE / SP-CVR	Unstable action switching

Figure 12 presents representative successful and failure cases in a nine-panel layout. The first row shows successful cases, where sensor signals, retrieved clauses, predicted actions, and protocol-reference actions are generally consistent. The second row illustrates boundary RUL failures, where samples lie near transitions between healthy and degraded states, causing unstable switching among monitor, warning, and inspect. The third row shows retrieval failures, where retrieved clauses are not fully aligned with the current equipment state, leading to constraint-signal deviation and actions closer to baseline predictions.

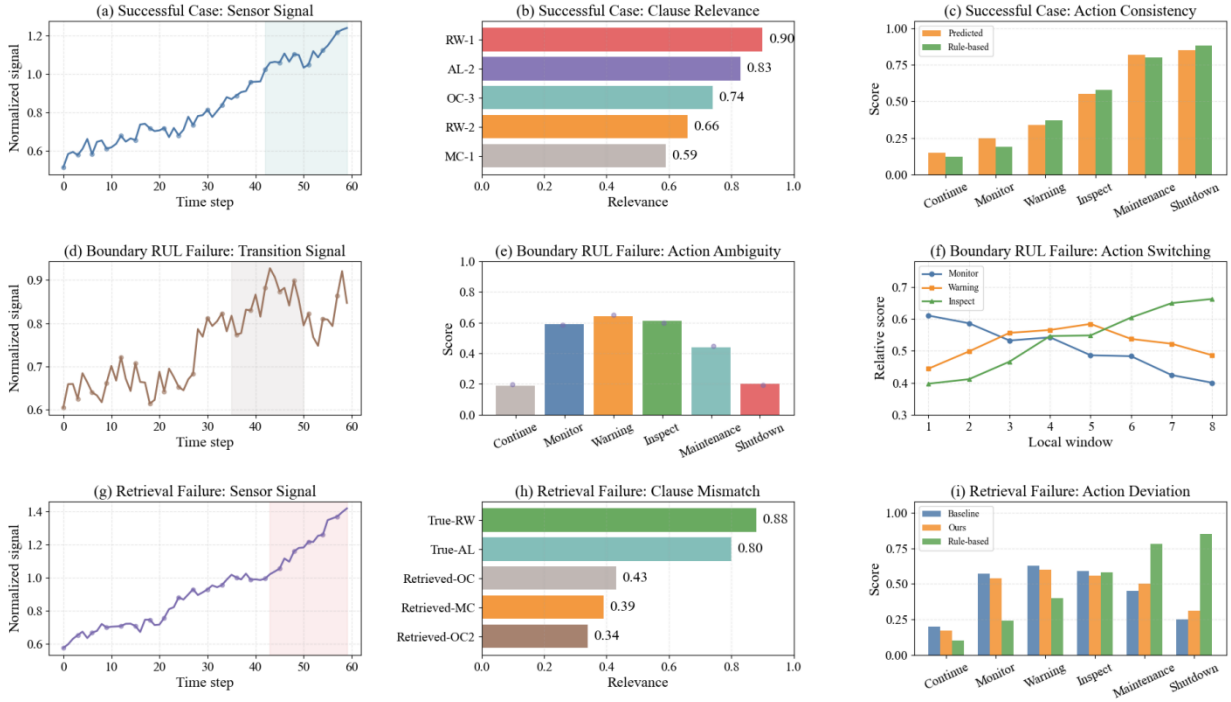

Figure 12. Successful and failure case analysis: (a)–(c) Successful cases; (d)–(f) Boundary RUL failures; (g)–(i) Retrieval failures

Overall, failures are mainly concentrated in semantic retrieval errors and boundary-state uncertainty. This indicates that ELIC depends on the quality and retrieval accuracy of English technical clauses, while HFLO remains limited in handling conflicting feedback signals. These observations are consistent with the SP-CVR and SP-FAS trends reported above and suggest future improvements in industrial semantic retrieval, clause-confidence estimation, and feedback-uncertainty modeling.

4.7 Ablation Study

To separate module effects from backbone capacity and information access, shared-backbone controls and component ablations are conducted on MetroPT-3 and XJTU-SY. All controlled variants use the same TCN–Transformer backbone, hidden dimensions, input windows, data splits, optimizer, training epochs, random seeds, task-to-action mapping, and external evaluator. They differ only in access to clauses, clause-derived supervision, feedback, and the corresponding objectives.

The shared-backbone-only variant uses equipment states without documents or feedback. The RAG-style variant accesses retrieved clauses but not clause-derived labels, ELIC loss, or feedback. HFLO-only and ELIC-only correspond to removing ELIC and HFLO, respectively. Component variants further replace relevant clauses with random English clauses or remove risk feedback, preference feedback, or constraint-aware inference.

As shown in table 11, all results are reported as mean ± standard deviation over five runs. HFLO-related variants use the independent risk–action protocol in Section 4.1.1, which does not access retrieved clauses, ELIC scores, or clause-derived labels.

Table 11. Ablation results on two datasets

Group	Variant	MetroPT-3 F1 ↑	MetroPT-3 SP-CVR ↓	MetroPT-3 SP-FAS ↑	XJTU-SY RMSE ↓	XJTU-SY SP-CVR ↓	XJTU-SY SP-FAS ↑
Controlled	Shared backbone only	0.887±0.004	0.174±0.010	0.738±0.011	12.56±0.18	0.176±0.009	0.724±0.011
Controlled	RAG-style clauses	0.879±0.006	0.151±0.008	0.762±0.010	13.08±0.21	0.149±0.008	0.751±0.010
Controlled	HFLO only (w/o ELIC)	0.884±0.007	0.166±0.011	0.776±0.012	13.02±0.24	0.162±0.011	0.760±0.012
Controlled	ELIC only (w/o HFLO)	0.887±0.006	0.145±0.010	0.736±0.015	12.90±0.22	0.141±0.010	0.728±0.014
Controlled	Full model	0.895±0.004	0.124±0.007	0.798±0.008	12.68±0.17	0.121±0.007	0.785±0.008
Component	Random English clauses	0.886±0.007	0.171±0.012	0.766±0.013	13.07±0.24	0.166±0.012	0.758±0.013
Component	w/o risk feedback	0.890±0.006	0.152±0.010	0.759±0.013	12.86±0.21	0.149±0.010	0.746±0.013
Component	w/o preference feedback	0.891±0.006	0.137±0.009	0.751±0.014	12.81±0.20	0.134±0.009	0.739±0.014
Component	w/o constraint-aware inference	0.889±0.007	0.155±0.010	0.765±0.013	12.94±0.23	0.154±0.010	0.756±0.013

Under the shared backbone, RAG-style clause access reduces SP-CVR from 0.174 to 0.151 on MetroPT-3 and from 0.176 to 0.149 on XJTU-SY, but remains less effective than explicit ELIC supervision. The full model achieves the lowest SP-CVR on both datasets, while the shared backbone retains the lowest XJTU-SY RMSE. HFLO-only yields higher SP-FAS than ELIC-only, whereas ELIC-only produces larger SP-CVR reductions. These results indicate metric-specific contributions rather than comprehensive superiority.

Replacing relevant clauses with random English clauses increases SP-CVR and reduces SP-FAS, showing that performance depends on state- and action-related clause semantics rather than generic text. Removing risk feedback causes a larger SP-CVR increase, whereas removing preference feedback produces a larger SP-FAS decrease. Thus, risk feedback mainly discourages insufficiently conservative actions, while preference feedback regulates agreement with the simulated preference protocol.

Removing constraint-aware inference changes task metrics only slightly but degrades both SP-CVR and SP-FAS, indicating that training-stage document and feedback integration alone is insufficient for final action ranking. Figures 13 and 14 visualize the component-ablation changes relative to the full model.

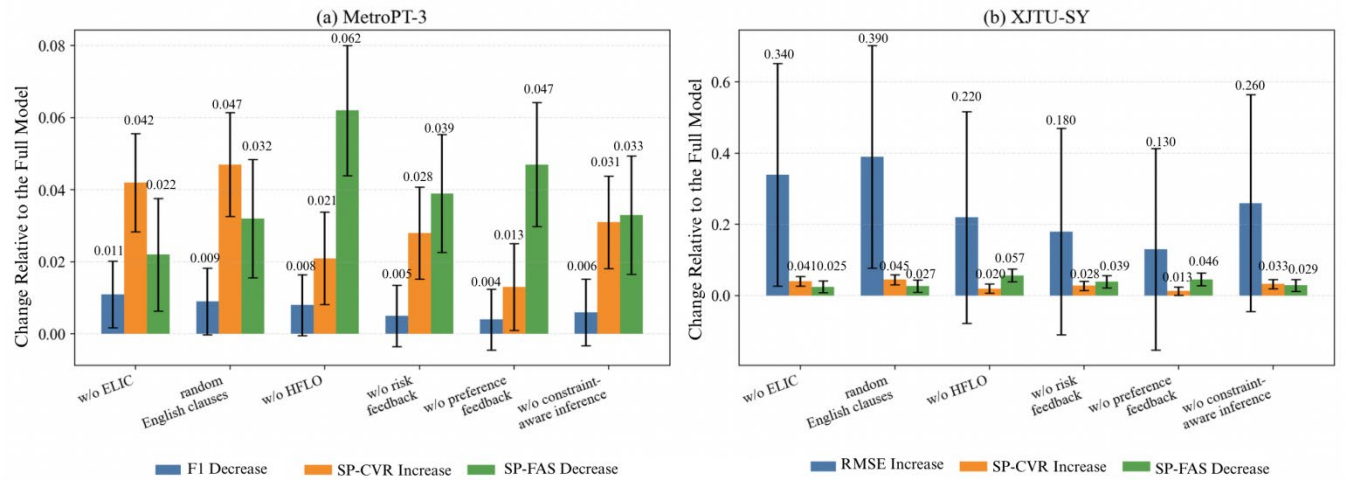

Figure 13. Metric changes of selected ablation variants relative to the full model: (a) F1 decrease, SP-CVR increase, and SP-FAS decrease on MetroPT-3; (b) RMSE increase, SP-CVR increase, and SP-FAS decrease on XJTU-SY

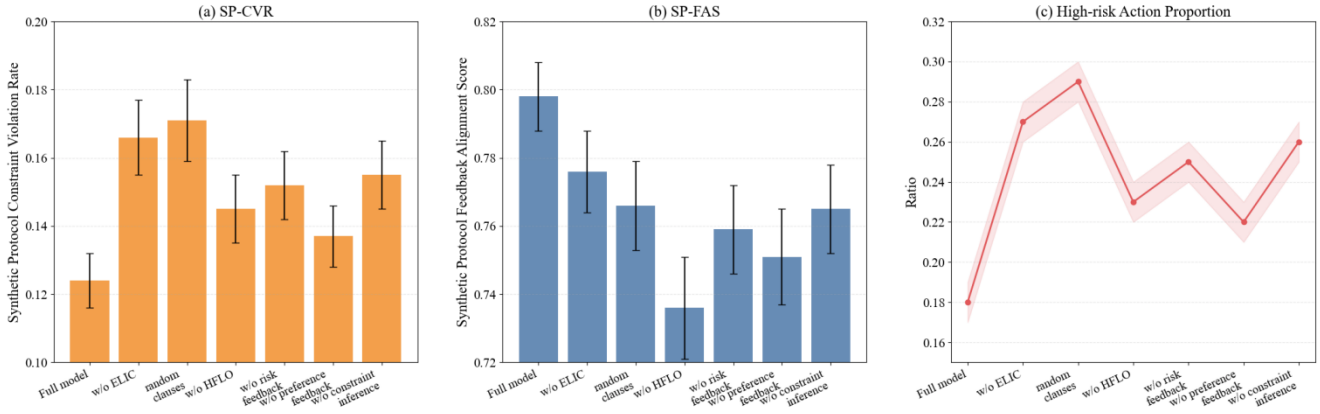

Figure 14. Behavioral changes for selected ablation variants: (a) SP-CVR, (b) SP-FAS, and (c) high-risk action proportion

Overall, the shared-backbone controls reduce confounding from model capacity. ELIC mainly improves document-feasibility learning and synthetic constraint consistency, HFLO improves feedback-protocol consistency, and constraint-aware inference integrates both signals during final action selection.

5. Discussion

This paper proposes an English technical document-enhanced human-in-the-loop maintenance decision-making method. Different from predictive maintenance approaches that rely solely on sensor-based prediction, this work does not directly equate fault probabilities or RUL estimation with maintenance actions. Unlike methods that primarily treat documents as retrieval context or explanatory material, the proposed approach uses ELIC to represent operating conditions, action limitations, and risk warnings from English equipment manuals, SOPs, and safety instructions as state–document – action feasibility signals, and incorporates preference and risk feedback into the optimization process through HFLO.

ELIC estimates document feasibility scores based on equipment state, candidate actions, and English clauses, enabling English technical documents to participate in both training loss computation and inference-stage action ranking. HFLO introduces feedback signals, allowing risk feedback to impose stronger constraints when document feasibility is low. The constraint-aware inference mechanism integrates task prediction scores, document feasibility, and feedback consistency, thereby avoiding reliance on a single prediction signal for maintenance decisions.

Under the controlled evaluation protocol constructed using MetroPT-3 and XJTU-SY, the proposed method achieves the lowest SP-CVR on both datasets and the highest F1 on MetroPT-3. However, it does not achieve the lowest RMSE or highest SP-FAS on XJTU-SY. Its main observed advantage therefore lies in reducing violations under the constructed clause protocol while maintaining task performance close to strong temporal baselines. Although the proposed method is not optimal on all individual metrics, it demonstrates stable performance in reducing English technical clause constraint violations while maintaining competitive predictive performance compared with strong temporal models. Representation analysis and evidence visualization provide auxiliary evidence that English clause semantics may improve action feasibility representations and offer traceable support for maintenance decision-making.

The proposed method is suitable for maintenance decision-support scenarios where English equipment manuals, maintenance regulations, and operational feedback are available. Its goal is not to replace human maintenance engineers, but to provide a computable relationship among predictive results, technical documents, and feedback information.

This study still has several limitations. First, although feedback generation is separated from document-derived constraints, the maintenance actions and operator-like feedback remain rule-based rather than field-observed. The independent risk–action protocol reduces direct overlap between ELIC and HFLO supervision but cannot fully represent differences in operator experience, preferences, or risk perception. Therefore, SP-CVR and SP-FAS reflect relative consistency with the constructed synthetic protocol rather than field-observed maintenance quality or actual operator judgments. Future work will collect expert feedback independently of clause annotation and validate the framework using field maintenance records. Second, the two datasets involve limited equipment types and task settings and may not fully represent all industrial maintenance scenarios. Third, ELIC depends on the quality of English clause retrieval, and HFLO has limited capability in handling feedback conflicts and uncertainty. Future work will focus on constructing higher-quality English maintenance clause libraries, incorporating real expert feedback, conducting cross-equipment, cross-condition, and online maintenance experiments, and evaluating model inference cost and deployment constraints.

6. Conclusion

To address the decoupling among equipment state prediction, English technical document utilization, and feedback optimization in predictive maintenance, this paper proposes an English technical document-enhanced human-in-the-loop maintenance decision-making framework. The proposed framework uses ELIC to represent maintenance conditions, action restrictions, and risk warnings from English equipment manuals, SOPs, and safety instructions as state–document–action feasibility signals, and incorporates preference and risk feedback into a joint optimization process via HFLO, thereby enabling document-constrained and feedback-consistent maintenance action modeling.

Experimental results under the controlled protocol show that the proposed method achieves the lowest SP-CVR on both datasets and the highest F1 on MetroPT-3, while TCN–Transformer retains a lower RMSE and the human-in-the-loop baseline retains a higher SP-FAS on XJTU-SY. The results support lower synthetic clause-protocol violation rates and traceable clause–feedback–action evidence within the constructed evaluation setting, rather than comprehensive superiority or field-validated decision improvement. Ablation studies further show that ELIC, English technical clause semantics, HFLO, and constraint-aware inference all contribute to the full model.

Future work will further explore higher-quality English maintenance clause construction, incorporation of real expert feedback, modeling of feedback uncertainty, and validation in cross-equipment, cross-condition, and online maintenance scenarios. In addition, model inference cost and practical deployment conditions will be evaluated to clarify its application boundaries in industrial maintenance systems.

References

[1] Fathi, K., Ristin, M., Sadurski, M., Kleinert, T., & Van De Venn, H. W. (2024, June). Detection of novel asset failures in predictive maintenance using classifier certainty. In 2024 32nd Mediterranean Conference on Control and Automation (MED) (pp. 50-56). IEEE.

[2] Fathi, K., Kleinert, T., & van de Venn, H. W. (2024, June). Trustworthy machine learning operations for predictive maintenance solutions. In PHM Society European Conference (Vol. 8, No. 1, pp. 4-4).

[3] Bhattacharya, M., Penica, M., O'Connell, E., Southern, M., & Hayes, M. (2023). Human-in-loop: A review of smart manufacturing deployments. Systems, 11(1), 35.

[4] Yanytska, L. (2025). The rise of human-centric manufacturing in the industry 5.0 era. The International Journal of Advanced Manufacturing Technology, 139(9), 5067-5077.

[5] Amaliah, N. R., Tjahjono, B., & Palade, V. (2025). Human-in-the-Loop XAI for Predictive Maintenance: A Systematic Review of Interactive Systems and Their Effectiveness in Maintenance Decision-Making. Electronics, 14(17), 3384.

[6] Chen, H., Li, S., Fan, J., Duan, A., Yang, C., Navarro-Alarcon, D., & Zheng, P. (2025). Human-in-the-loop robot learning for smart manufacturing: A human-centric perspective. IEEE Transactions on Automation Science and Engineering, 22, 11062-11086.

[7] Moosavi, S., Farajzadeh-Zanjani, M., Razavi-Far, R., Palade, V., & Saif, M. (2024). Explainable AI in manufacturing and industrial cyber–physical systems: A survey. Electronics, 13(17), 3497.

[8] Bayat, M., & Kharel, S. (2025). Leveraging Artificial Intelligence for Predictive Maintenance and Condition Rating of Off-System Bridges. Applied Sciences, 15(21), 11301.

[9] Lantu, D. C., Lestari, Y. D., & Putri, A. N. A. (2026). Human-in-the-Loop: From Complete Automation to Dark Factories. Foresight and STI Governance, 20(2).

[10] Gómez Fernández, J. F., & Crespo Márquez, A. (2026). Artificial Intelligence and Machine Learning in Industrial Maintenance Optimization. In Digital Maintenance and Asset Digitalization: Smart Strategies for Optimizing Asset Performance (pp. 119-143). Cham: Springer Nature Switzerland.

[11] Siraskar, R., Kumar, S., Patil, S., Bongale, A., & Kotecha, K. (2023). Reinforcement learning for predictive maintenance: A systematic technical review. Artificial Intelligence Review, 56(11), 12885-12947.

[12] Yang, M., Su, Y., Lei, Z., Deng, S., Zhang, Z., & Wen, G. (2025). Digital twin-driven predictive maintenance methods for the critical components of rotating machinery: A review of the research. Equipment Intelligent Operation and Maintenance, 289-296.

[13] Bukowski, L., & Werbinska-Wojciechowska, S. (2025). Towards maintenance 5.0: resilience-based maintenance in AI-driven sustainable and human-centric industrial systems. Sensors, 25(16), 5100.

[14] Noot, J. P., Martin, M., & Birmele, E. (2025). Lstm and transformers based methods for remaining useful life prediction considering censored data. International Journal of Prognostics and Health Management, 16(2).

[15] Raffik, R., Maria, W. A., Subashini, B., & Asvitha, R. (2025). Artificial Intelligence-Based Predictive Maintenance Approaches for Vehicle Condition Monitoring and On-Board Diagnostic Systems to Enhance Automotive Industries. In Industry 5.0 for Society 5.0: Revolutionizing Smart Farming, Manufacturing, and Green Computing (Part 2) (pp. 107-137). Bentham Science Publishers.

[16] Wang, B., Zheng, P., Song, C., Mourtzis, D., & Wang, L. (2025). Future research directions on human-centric smart manufacturing. Human-Centric Smart Manufacturing Towards Industry 5.0, 359-369.

[17] Converso, G., Gallo, M., Murino, T., & Vespoli, S. (2023). Predicting failure probability in Industry 4.0 production systems: a workload-based prognostic model for maintenance planning. Applied Sciences, 13(3), 1938.

[18] Suci, A. M., Amini, R., Asri, A. K., & Martin, N. (2025). Artificial intelligence in renewable energy: A review of predictive maintenance and energy optimization. Journal of Clean Technology, 2(1), 29-44.

[19] Sherif, Z., & Salonitis, K. (2025). A systematic review of decision tools for process selection and performance improvement in manufacturing. The International Journal of Advanced Manufacturing Technology, 1-29.

[20] Jiang, M., Jiang, T., Guo, L., & Liu, S. (2025). A Scheduling Method for Maintenance Tasks of Damaged Equipment Based on Digital Twin and Robust Optimization. Sensors, 25(18), 5674.

[21] Van Dinter, R., Tekinerdogan, B., & Catal, C. (2023). Reference architecture for digital twin-based predictive maintenance systems. Computers & Industrial Engineering, 177, 109099.

[22] Shamim, M. M. R. (2025). Maintenance optimization in smart manufacturing facilities: A systematic review of lean, TPM, and digitally-driven reliability models in industrial engineering. American Journal of Interdisciplinary Studies, 6(1), 144-173.

[23] Orošnjak, M., Saretzky, F., & Kedziora, S. (2025). Prescriptive maintenance: A systematic literature review and exploratory meta-synthesis. Applied Sciences, 15(15), 8507.

[24] Abdullahi, I., Longo, S., & Samie, M. (2024). Towards a distributed digital twin framework for predictive maintenance in industrial internet of things (IIoT). Sensors, 24(8), 2663.

[25] Allen, L., Lu, H., & Cordiner, J. (2024). Knowledge-enhanced spatiotemporal analysis for anomaly detection in process manufacturing. Computers in Industry, 161, 104111.

[26] Ucar, A., Karakose, M., & Kırımça, N. (2024). Artificial intelligence for predictive maintenance applications: key components, trustworthiness, and future trends. Applied Sciences, 14(2), 898.

[27] Giacotto, A., Marques, H. C., & Martinetti, A. (2025). Prescriptive maintenance: a comprehensive review of current research and future directions. Journal of quality in maintenance engineering, 31(1), 129-173.
[28] Hong, S., Shin, J. M., Seo, J., Lee, T., Park, J., Young, C. M., ... & Lim, H. S. (2024, November). Intelligent predictive maintenance RAG framework for power plants: Enhancing QA with StyleDFS and domain specific instruction tuning. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track (pp. 805–820).
[29] Turner, C., Okorie, O., & Oyekan, J. (2022). XAI sustainable human in the loop maintenance. IFAC-PapersOnLine, 55(19), 67–72.
[30] Nikitin, A., & Kaski, S. (2022, August). Human-in-the-loop large-scale predictive maintenance of workstations. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 3682–3690).
[31] Alexander, Z., Chau, D. H., & Saldaña, C. (2024). An interrogative survey of explainable AI in manufacturing. IEEE Transactions on Industrial Informatics, 20(5), 7069–7081.
[32] Pagan, N., Baumann, J., Elokda, E., De Pasquale, G., Bolognani, S., & Hannák, A. (2023, October). A classification of feedback loops and their relation to biases in automated decision-making systems. In Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (pp. 1–14).
[33] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.
[34] Chen, T., & Guestrin, C. (2016, August). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794).
[35] Graves, A. (2012). Long short-term memory. In Supervised Sequence Labelling with Recurrent Neural Networks (pp. 37–45).
[36] Chen, D., Hong, W., & Zhou, X. (2022). Transformer network for remaining useful life prediction of lithium-ion batteries. IEEE Access, 10, 19621–19628.
[37] Fan, Z., Li, W., & Chang, K. C. (2024). A two-stage attention-based hierarchical transformer for turbofan engine remaining useful life prediction. Sensors, 24(3), 824.
[38] Jin, X., Ji, Y., Li, S., Lv, K., Xu, J., Jiang, H., & Fu, S. (2025). Remaining useful life prediction for rolling bearings based on TCN–Transformer networks using vibration signals. Sensors, 25(11), 3571.