mathematics

MDPI

Article

Formal Modeling of Augmented Reality Maintenance-Guidance Processes Based on Multimodal Knowledge Extraction and Graph Reasoning

Hongduo Wu [1,2], Xingzheng Wei [1,2], Yifu Yan [1,2], Ziyue Guo [1,3,*] and Dong Zhou [1,2]

[1] State Key Laboratory of Reliability and Environmental Engineering Technology, Beihang University, Beijing 100191, China; wuhongduo@buaa.edu.cn (H.W.)
[2] School of Reliability and System Engineering, Beihang University, Beijing 100191, China
[3] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
* Correspondence: guoziyue@buaa.edu.cn

Abstract

Augmented reality maintenance guidance (ARMG) provides support for industrial maintenance by converting knowledge into instructions and integrating virtual information with physical equipment. However, constructing guidance processes still depends on the manual interpretation of heterogeneous maintenance materials, while inconsistent knowledge representations and missing information hinder automated modeling. This study proposes a formal modeling method that integrates multimodal knowledge extraction with graph reasoning. A Maintenance Guidance Ontology (MGO) and a six-tuple maintenance-event template are first developed to unify the semantic representation of actions, objects, resources, and related elements. Qwen3-VL is then fine-tuned to the ARMG domain through low-rank adaptation (LoRA) to extract maintenance knowledge from text and video, respectively. Finally, for missing elements in extracted knowledge, we develop an MGO-constrained semantic–temporal graph-convolutional network (MGO-STGCN). It combines relation semantics, step order, weighted neighborhood aggregation, and an ontology-constrained mask. The model predicts and completes three missing-element types: tools, spare parts and consumables, and precautions. Both extraction models outperformed the comparison methods in terms of slot-level and event-level F1 scores. MGO-STGCN achieved F1 scores of 0.799, 0.735 and 0.680 for the three completion tasks. Ablation experiments indicated that relation–temporal weighting and ontological constraints make complementary contributions to completion performance. These results demonstrate the potential of the proposed framework for unified formal representation of multimodal maintenance knowledge and completion of missing guidance information.

Keywords: augmented reality maintenance guidance; ontology modeling; large vision-language model; multimodal knowledge extraction; graph neural network

MSC: 68T07; 68T30; 68T40

Check for updates

Academic Editor: Iraklis Varlamis

Received: 15 October 2026
Revised: 4 November 2026
Accepted: 5 November 2026
Published: 9 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Modern complex aeronautical equipment comprises large structures and tightly coupled electromechanical systems, so maintenance support requires extensive specialist knowledge and operating procedures [1]. Maintenance personnel therefore face substantial cognitive load and a high risk of human error [2]. They must still retrieve, compare, and verify

task information from aircraft maintenance manuals (AMMs), fault isolation manuals (FIMs), and job cards.

Augmented reality maintenance guidance (ARMG) presents intuitive instructions by integrating virtual information with physical equipment. This technology can improve maintenance efficiency, quality, and safety [3–5]. However, most existing guidance processes are developed for specific equipment and tasks [6,7] and lack a unified knowledge representation mechanism [8]. Consequently, maintenance knowledge is difficult to reuse across tasks, and process construction and updating still require substantial manual effort.

This study investigates how heterogeneous and incomplete maintenance materials can be transformed into structured, formally represented maintenance-event sequences. We propose a formal modeling method comprising three stages: knowledge representation, knowledge acquisition, and knowledge reasoning. The method aims to support the construction of reusable maintenance-guidance processes by extracting event information and completing missing elements.

The main contributions are as follows:

1. A Maintenance Guidance Ontology (MGO) and a six-tuple event representation are developed to unify the semantic relations among maintenance actions, objects, resources, constraints, and states. They provide a formal interface for cross-modal extraction and graph reasoning.
2. A maintenance-knowledge extraction method is developed for multimodal maintenance materials using a vision-language model fine-tuned with maintenance knowledge. A unified output structure constrains event generation from maintenance text and video. The text and video channels achieve macro-averaged F1 scores of 0.735 and 0.547, respectively.
3. A graph-convolutional network is proposed to reason about and complete missing elements in maintenance-guidance processes. It combines relation–temporal weighted graph convolution with ontology-mask-constrained candidate prediction. The resulting F1 scores are 0.799, 0.735 and 0.680 for tools, spare parts and consumables, and precautions, respectively.

The remainder of this paper reviews related work, presents the proposed method, describes the experimental design, discusses the results, and concludes the study.

2. Related Work

2.1. Formal Representation of Maintenance Knowledge

Recent advances in ontologies [9], knowledge graphs [10], and artificial intelligence have shifted related research from static rule modeling toward dynamic cognitive modeling. In knowledge engineering, this progression encompasses knowledge representation, knowledge acquisition, and knowledge reasoning [11].

For formal maintenance-knowledge representation, ontologies based on the Web Ontology Language (OWL) provide concept hierarchies, relationship constraints, and consistency checking. They have become a standard approach to structured knowledge representation. An ontology formally specifies a shared conceptualization by describing the object types, concepts, attributes, and relationships in a domain [9]. Lin et al. [12] developed an ontology for civil-aircraft structural maintenance cases to support knowledge retrieval and reuse. Chen et al. [13] constructed an ontology for a complex-equipment fault-diagnosis knowledge graph that provided a semantic framework for knowledge extraction. Huang et al. [14] proposed an ontology-guided approach to multi-level knowledge-graph construction and demonstrated its application to industrial process analysis. Feng et al. [15] developed a knowledge-graph ontology for early-warning equipment maintenance and support to

improve maintenance and resource-support efficiency. Collectively, these studies show a transition from static structural descriptions to semantic and context-aware knowledge representations.

2.2. Knowledge Extraction from Multimodal Maintenance Materials

Knowledge acquisition extracts and integrates machine-interpretable, reusable knowledge from heterogeneous multimodal maintenance materials. Its central task is the automatic extraction of structured knowledge from multiple data sources.

Maintenance-text extraction primarily uses sequence-labeling or semantic-generation methods. Chen et al. [13] proposed a BiGRU-Attention model that improved entity and relation extraction from complex-equipment fault texts. Kong et al. [16] combined prompt learning with BERT semantic similarity to extract fault-diagnosis knowledge from aircraft-engine maintenance records. BERT-BiLSTM-CRF sequence labeling has also become an important paradigm. Xu et al. [17] employed an active-learning-based BERT–BiLSTM–CRF model to extract fault knowledge from construction machinery maintenance records. Geng et al. [18] applied it to named-entity recognition in Chinese vehicle-maintenance cases. Wei et al. [19] improved the recognition of power-equipment maintenance work orders by combining position-aware and similarity-aware attention. However, BERT-BiLSTM-CRF methods mainly address token-level or short-sentence labeling and have limited capacity to represent dispersed entities, attributes, and rule-based knowledge in aircraft maintenance manuals. Such manuals also contain heterogeneous complementary information in figures and tables. Omitted subjects, fronted conditions, nested hierarchies, and cross-step references are common, making complete event structures difficult to recover using only contextual encoding and CRF label transitions. Large language models (LLMs) use Transformer architectures, large-scale pretraining, and task fine-tuning to provide strong language understanding and generation in specialized domains. Dong et al. [20] used prompt engineering and few-shot strategies with an LLM to extract and align mine-ventilation information. Giordano and Fantoni [21] used an LLM to analyze fault descriptions and maintenance actions in work orders and decomposed those actions into subtasks. Li et al. [22] combined knowledge extraction, graph attention networks, and retrieval-augmented generation for wind-power-equipment maintenance. Their results indicate that integrating LLMs with knowledge graphs and graph neural networks can improve the structured use of complex maintenance knowledge.

For maintenance images and videos, deep-learning models have substantially advanced the semantic analysis of maintenance actions. Vico et al. [23] used a three-dimensional convolutional neural network to classify industrial-assembly actions online and recognize action sequences. A multiscale central-difference graph-convolutional network [24] and a hierarchical intertwined graph-learning framework [25] improved skeleton-based action recognition using sparse self-attention and temporal Transformers, respectively. Vision-language models (VLMs) provide a further route to knowledge extraction from maintenance videos. Deng et al. [26] used a VLM to extract operating steps, components, and spatial relations from disassembly videos for generating AR repair instructions. Ren et al. [27] proposed TimeChat to improve temporal localization and understanding in long videos. Nevertheless, these methods mainly produce action labels or natural-language descriptions. Their outputs remain inconsistent with the unified event slots and process relations required for maintenance guidance.

2.3. Reasoning and Completion of Maintenance Knowledge

Maintenance-knowledge reasoning and completion generally rely on rule-based reasoning or graph reasoning over knowledge graphs. Traditional Semantic Web Rule Language (SWRL) reasoning provides explicit and interpretable inference rules, but its flexibility is

limited when handling incomplete information and differences in the importance of rule antecedents [28]. Knowledge-graph-based completion has therefore attracted increasing attention. Kong et al. [16] developed an aircraft-engine knowledge graph with fault-diagnosis reasoning based on semantic similarity. Current research further shows that combining knowledge graphs with graph neural networks (GNNs) provides an effective means of modeling associations and completing complex maintenance knowledge. Yang et al. [29] proposed a hybrid semantic-structural GNN for completing fault-knowledge graphs in railway equipment. Xia et al. [30] constructed a maintenance-knowledge graph for planning recommendations and proposed an attention-compressed relational graph-convolutional network to predict potential fault causes and maintenance plans. These studies focus mainly on fault causes, maintenance plans, and their associations. They give comparatively limited attention to the temporal relations and resource constraints among individual maintenance steps.

Existing studies have advanced the representation, acquisition, and reasoning of maintenance knowledge, but three limitations remain. First, dedicated ontologies for ARMG processes are lacking. Existing ontologies generally focus on fault diagnosis and provide limited support for standardized process descriptions and guidance semantics. Second, current extraction methods do not consistently interpret specialist terminology, operating intent, and cross-modal process semantics, leading to ambiguous or variable outputs. Third, maintenance-knowledge-graph reasoning concentrates on fault diagnosis and rarely completes missing elements in maintenance-operation sequences.

To address these limitations, we propose a formal modeling framework for ARMG processes. The framework uses the MGO to standardize maintenance-event semantics and VLMs fine-tuned on maintenance-domain knowledge to extract structured events from maintenance text and videos. An ontology-constrained graph neural network then predicts missing maintenance tools, spare parts and consumables, and precautions.

3. Methods

3.1. Overall Framework

To address the preceding problems, we developed a method for formally modeling ARMG processes from heterogeneous maintenance materials. Figure 1 shows the workflow. First, the MGO standardizes maintenance-event concepts and relations and defines a six-tuple maintenance-event template. Second, low-rank adaptation (LoRA) and the Qwen3-VL large vision-language model are used to train separate extraction models for maintenance text and video. These models, denoted M1 and M2, efficiently convert multimodal maintenance materials into structured guidance knowledge. Finally, the extracted knowledge is mapped to a maintenance-guidance-process knowledge graph. An MGO-constrained semantic–temporal graph-convolutional network (MGO-STGCN) predicts and fills missing elements to generate structured process sequences for ARMG.

3.2. Maintenance Guidance Ontology and Maintenance-Event Representation

We first developed an ontology dedicated to ARMG processes. The MGO provides a unified semantic interface for subsequent knowledge extraction and reasoning. During conceptual analysis, core guidance-process concepts were identified in the fault and maintenance domains. Fault knowledge comprises failure modes, causes, positions, and attributes. Maintenance knowledge comprises procedures, steps, personnel, resources, and maintenance criteria. Domain knowledge from AMMs, FIMs, and maintenance videos was used to define structural and operational-semantic relationships among these concepts. The resulting fault and maintenance concept models are shown in Figure 2 and provide explicit domain concepts and intuitive relation representations for constructing the MGO.

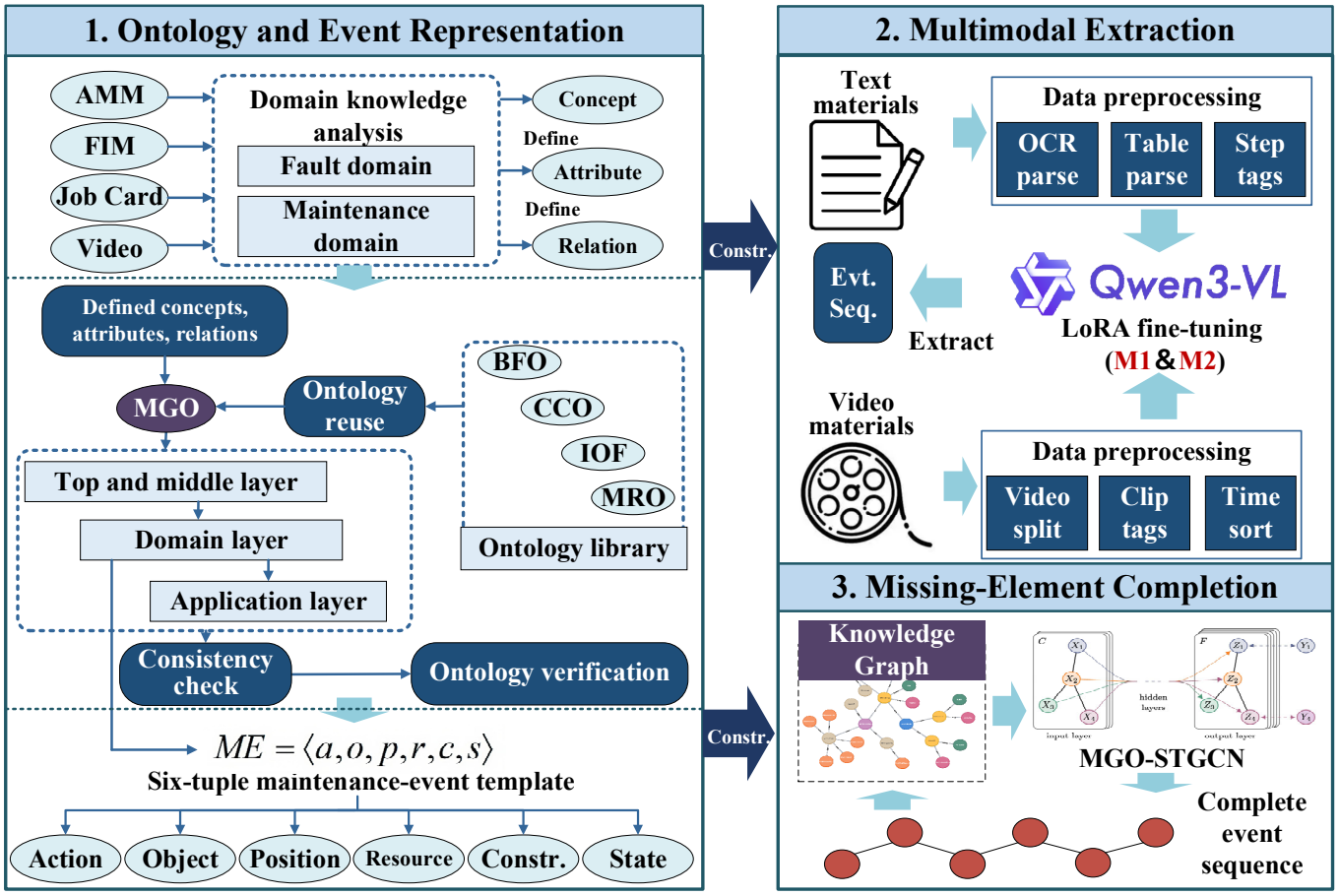


Figure 1. Workflow of the proposed formal modeling method for ARMG processes based on multimodal knowledge extraction and graph reasoning.

During ontology construction, the MGO reused relevant concepts and relations from four established resources. These were the Basic Formal Ontology (BFO) [31], Common Core Ontologies (CCO) [32], Industrial Ontologies Foundry (IOF) [33], and Maintenance Reference Ontology (MRO) [34]. The MGO is represented by the following five-tuple:

$$\mathrm{MGO} = \{E, A, R, P, C\} \tag{1}$$

Here, E denotes the set of entity-related concepts, and A denotes the set of attribute-related concepts. R contains non-hierarchical semantic relations among entities and attribute concepts. P contains hierarchical structural relations not included in R. C denotes the set of constraint axioms for guidance information associated with a specific maintenance object.

The MGO contains 54 classes, 15 object properties, 18 data properties, 39 explicit hierarchical relations, and 38 axiom rules. The logical consistency of the MGO was verified using the Pellet 2.3.1 in Protégé 5.5.0. No logical inconsistencies or unsatisfiable named classes were detected, confirming the logical consistency of the ontology.

The MGO comprises four layers: fault knowledge, maintenance process, resources and personnel, and guidance information, as shown in Figure 3.

The fault-knowledge layer describes the triggering conditions and semantic basis of faults in an ARMG process. It mainly includes failure modes, causes, attributes, and product-structure hierarchies. Relations such as “HasFailureMode” and “LeadTo” connect fault information with system functions, providing an ontological basis for inferring maintenance objects and requirements from fault information.

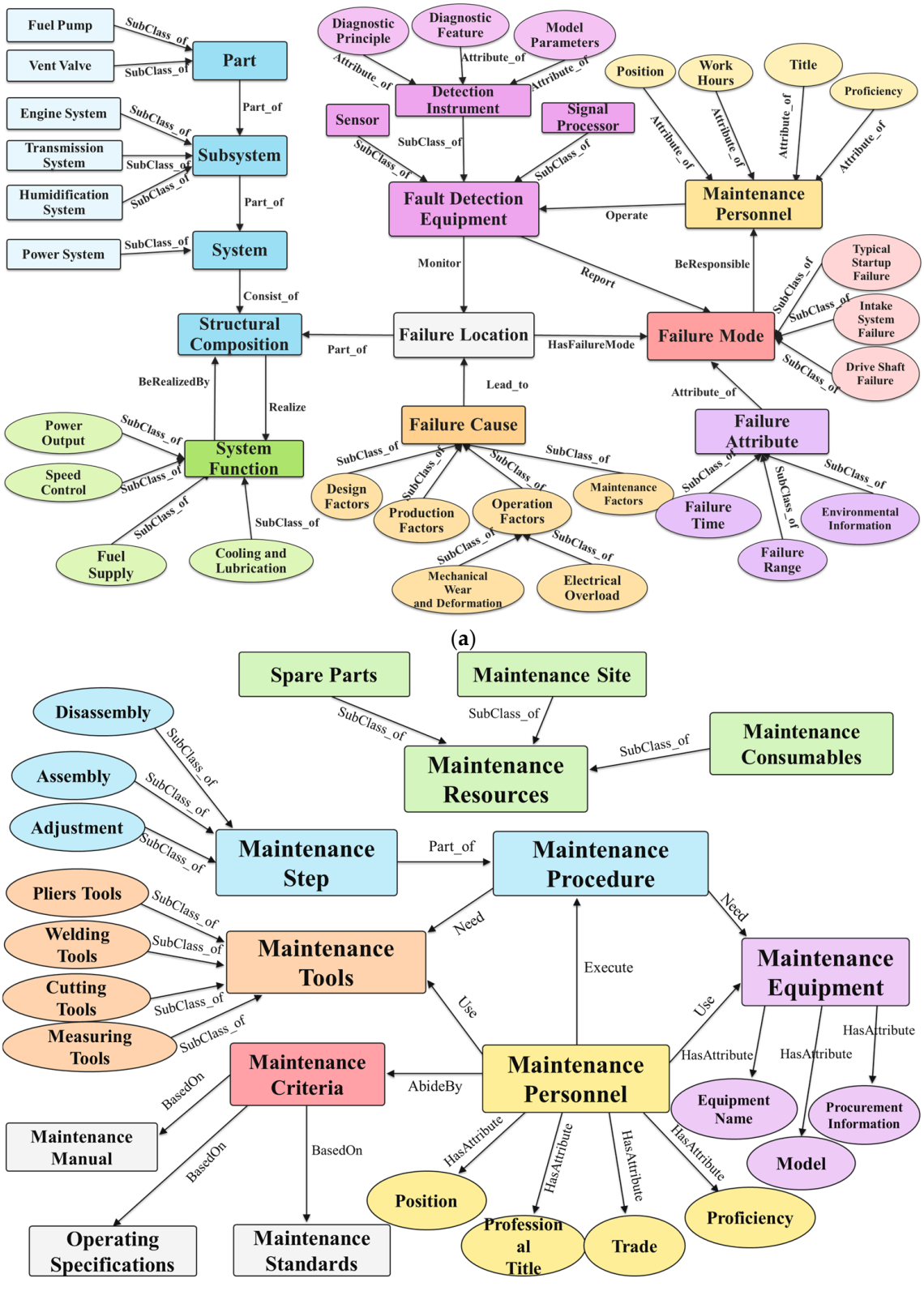


Figure 2. Conceptual and semantic-relation models for the fault and maintenance domains: (**a**) fault-related concepts and semantic relations and (**b**) maintenance-related concepts and semantic relations. (Arrows indicate the direction of semantic relationships between concepts).

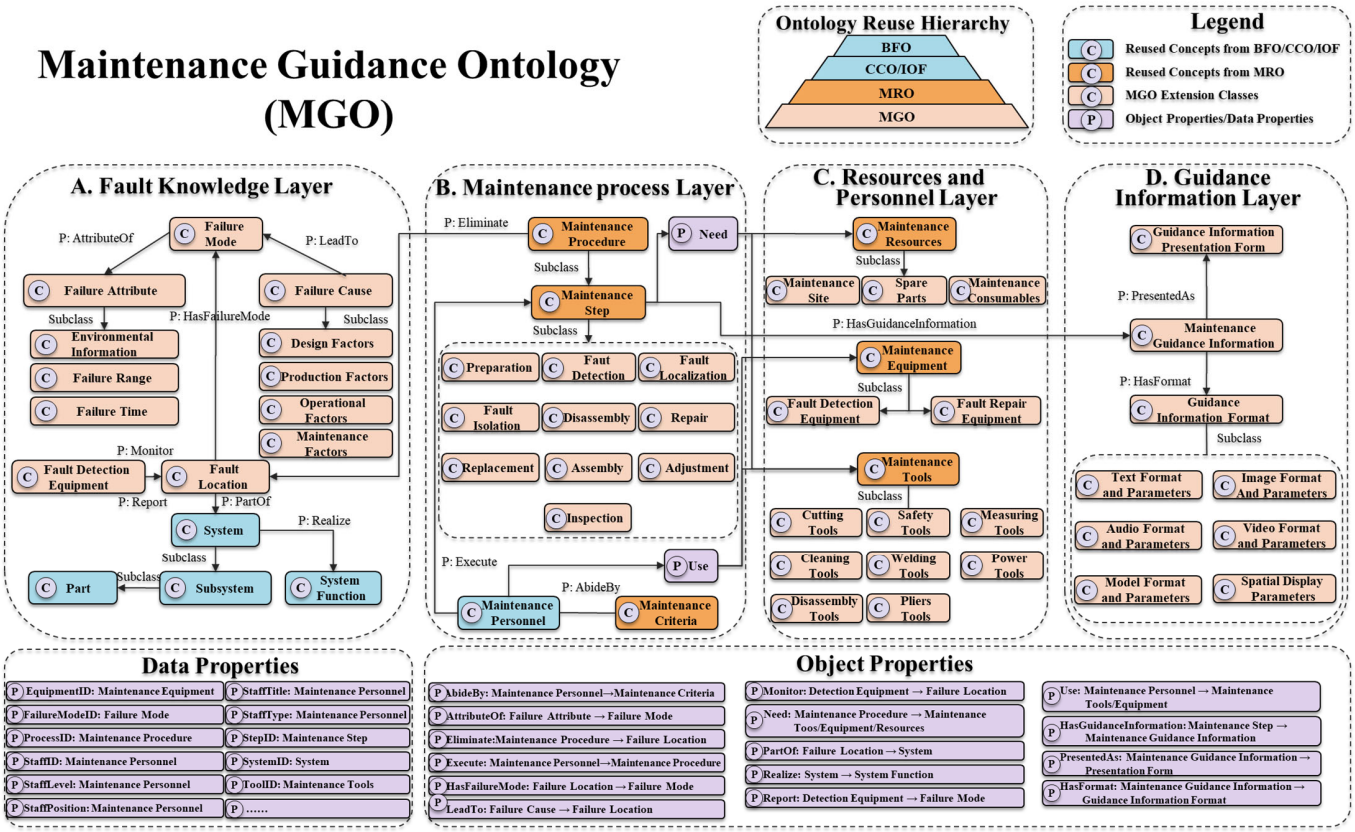


Figure 3. Structure of the Maintenance Guidance Ontology.

The maintenance-process layer describes procedural knowledge from fault treatment to functional recovery. It includes operation types such as fault detection, disassembly, and replacement, together with maintenance criteria. Relations such as "Eliminate", "Execute", and "AbideBy" represent maintenance objectives, executing agents, and procedural constraints.

The resources-and-personnel layer describes the supporting elements required to execute an ARMG process. It includes tools, equipment, personnel, sites, spare parts, and consumables. Relations such as "Need" and "Use" represent resource requirements and usage.

The guidance-information layer describes how maintenance-process knowledge is presented on an AR terminal and specifies the corresponding information formats. "HasGuidanceInformation" and "PresentedAs" connect maintenance steps with presentation forms such as text, speech, and images.

The MGO structure standardizes the semantic associations among fault information, maintenance activities, maintenance resources, and guidance information. It provides a unified schema for event extraction, knowledge-graph construction, completion of missing knowledge, and the design of ARMG information.

To further evaluate the functional adequacy of the MGO, 10 competency questions (CQs) were formulated to cover four knowledge layers (presented in Appendix A (Table A1)): fault knowledge (CQ1–CQ2), maintenance processes (CQ3–CQ4), resources and personnel (CQ5–CQ7), and guidance information (CQ8–CQ10). Each CQ was translated into a SPARQL query and evaluated using representative test instances. For rule-dependent questions, the applicable SWRL rules were executed before querying the inferred relations. The query results were compared with predefined expected answers. All ten CQs were successfully answered, achieving a coverage rate of 100%. Together with the Pellet con-

sistency check, this evaluation demonstrates the logical consistency and functional applicability of the MGO for representing and reasoning about maintenance-guidance knowledge.

Using this ontology, an individual maintenance step can be abstracted as a maintenance event containing six elements:

$$\mathrm{ME} = \langle a, o, p, r, c, s \rangle \tag{2}$$

where a denotes the action type, o the target object, p the position of the operation on the target object, r the resource element, c the prerequisite constraint, and s the resultant state. The six-tuple template in Equation (2) is the central semantic unit used throughout this study. Step statements in text, operation segments in videos, and process nodes in the knowledge graph are all represented using this template.

3.3. Maintenance-Guidance Knowledge Extraction from Multimodal Materials

Maintenance-guidance knowledge is distributed across maintenance manuals, job cards, fault records, maintenance videos, and other multimodal sources. Differences in information granularity, representation, and temporal structure can produce inconsistent step granularity, conflicting object names, and broken event chains. We used Qwen3-VL-8B-Instruct with LoRA to construct separate text- and video-extraction models, M1 and M2, which map knowledge from both modalities to the same six-tuple maintenance-event representation.

3.3.1. Multimodal Data Preprocessing

(1) Maintenance-text preprocessing based on multi-model parsing

Maintenance manuals and job cards contain intermixed text, images, and tables. To construct training samples for LoRA, we parsed these materials jointly using PaddleOCR 2.10.0, the Contrastive Language–Image Pre-training (CLIP) model [35], and TableMaster. PaddleOCR first recognized text and layout regions in non-editable documents. CLIP then computed semantic similarities between text fragments and associated figures. Let A and B denote the feature vectors of an image and a text fragment, respectively. Their cosine similarity in the shared feature space is defined as follows:

$$\mathrm{Sim}(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} \tag{3}$$

Cosine similarity measures the semantic correspondence between image and text features, providing a basis for associating maintenance-step descriptions with relevant images. For tabular materials, TableMaster employs a dual-branch decoder to predict cell positions and table structures, thereby recovering maintenance steps, tools, spare parts, and their original organization. After terminology normalization and step validation, the extracted information was organized into Markdown or JSON records containing text, image identifiers, and maintenance-step information for subsequent model fine-tuning.

(2) Maintenance-video preprocessing based on Temporal Perceiver

To construct supervised training samples, continuous maintenance videos were divided into semantic segments corresponding to individual maintenance steps. Temporal Perceiver [36] was used to detect action boundaries during this training-data preparation stage. ResNet-50 first extracts frame-level visual features to form the following feature sequence:

$$F = \{f_1, f_2, \ldots, f_n\} \in R^{n \times d} \tag{4}$$

where n is the number of video frames and d is the dimension of each frame-level feature. Temporal Perceiver introduces boundary queries and context queries. Through cross-attention, these queries capture semantic discontinuities between adjacent operations and

semantic consistency within a continuous action. The high-dimensional frame sequence is compressed into a boundary-sensitive temporal representation, after which a boundary decoder predicts segmentation locations and confidence scores.

The original videos were divided into maintenance-operation clips at the predicted boundaries, and each clip was labeled with a semantic step description. Key frames were sampled, and target regions were annotated for actions, objects, maintenance positions, and tools or other resources, as illustrated in Figure 4. These annotations enabled the model to learn correspondences among video actions, spatial objects, and maintenance-event slots. Video segmentation, step descriptions, and key-frame annotations were used only for training-data preparation and were not required during test-time inference.

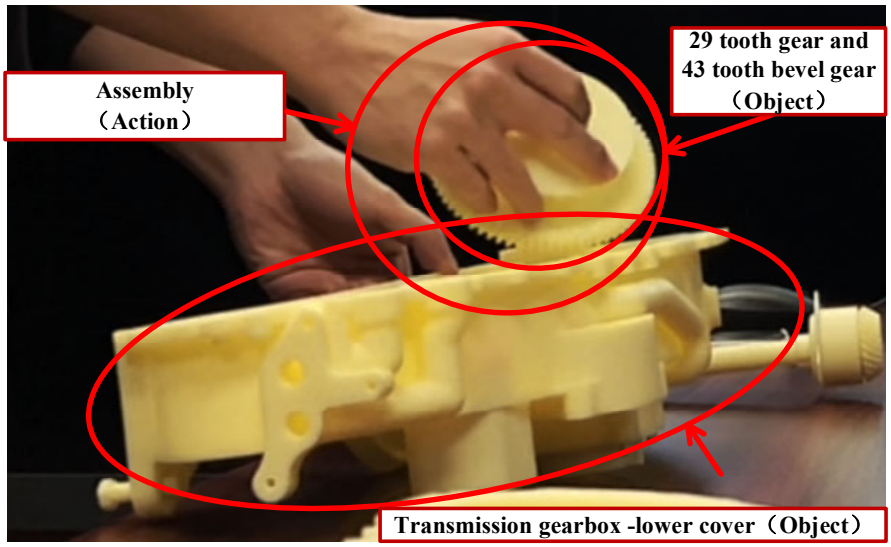


Figure 4. Example annotations for key-frame data.

After preprocessing, each text-training sample contains the original document, step text, image-text associations, and a six-tuple event label. Each video-training sample contains a video clip, a step description, target-region annotations, and a six-tuple event label. Both sample types use the same instruction-input–reference-output structure.

3.3.2. Domain-Knowledge-Enhanced Extraction of Maintenance-Guidance Processes

(1) Qwen3-VL and low-rank adaptation

Qwen3-VL-8B-Instruct was selected as the base model. Its visual encoder and multimodal joint-attention mechanism process maintenance documents, images, and videos, while its language decoder generates six-tuple maintenance events. The general model interpreted specialist terminology, step relations, and event slots inconsistently. LoRA instruction tuning therefore incorporated maintenance-domain knowledge and structured-output requirements into the model parameters.

LoRA freezes the base-model parameters and introduces trainable low-rank increments into selected linear mapping layers. Let $W_0 \in R^{d\times k}$ denote the original weight matrix of a linear Transformer layer and $X \in R^k$ its input. The output after introducing the parameter increment is

$$h = W_0X + \Delta WX \tag{5}$$

where ΔW is the learned increment matrix, referred to as the LoRA adapter. To reduce the number of trainable parameters, LoRA decomposes this matrix into two low-rank matrices:

$$A \in R^{r\times k},\ B \in R^{d\times r},\ \Delta W = BA \in R^{d\times k} \tag{6}$$

where $r \ll \min(d, k)$ is the low-rank dimension. After introducing the scaling factor α/r, the linear-layer output becomes

$$h = W_0 X + \frac{\alpha}{r} BAX \tag{7}$$

LoRA replaces direct learning of a high-dimensional weight increment with the product of two low-rank matrices. Parameter optimization is therefore performed in a low-dimensional subspace. This design adapts the model to maintenance terminology, action–object relations, and event-slot structures with limited parameter overhead.

The LoRA adapters were introduced into the query, key, value, and output projection matrices of the self-attention layers in the Qwen3-VL language decoder. They were also introduced into the up-projection, gate-projection, and down-projection layers of the feed-forward network. The model could therefore acquire task-specific knowledge for maintenance-event extraction without changing its main architecture.

(2) Structured knowledge extraction through conditional-probability optimization

Domain fine-tuning improves the model's representation of maintenance terminology and the six-tuple output structure, thereby reducing nonstandard terms, slot confusion, and unsupported completion. For the same maintenance description, a general model may generate different action expressions, object names, or supplementary details. A maintenance-guidance process instead requires a structured event with fixed slot definitions. We therefore use the reference six-tuple event as the supervision target and transform free generation into conditional optimization of the target sequence given the maintenance material.

Let θ denote the frozen Qwen3-VL parameters and $\Delta\theta$ the trainable LoRA parameters. For training sample i, P_i is the input instruction and $Y_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,T_i})$ is the target output sequence. Here, T_i is the number of tokens in the reference output, and $y_{i,t}$ denotes the token at position t in the i-th sample. For a domain-specific instruction dataset containing N training samples, the fine-tuning loss is defined as follows:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \left[\frac{1}{T_i} \sum_{t=1}^{T_i} \log p\left(y_{i,t} | P_i, y_{i,<t}; \theta + \Delta\theta\right) \right] \tag{8}$$

Figure 5 presents representative fine-tuning samples and inference prompts for M1 and M2. Both models share the MGO-defined six-tuple event schema and JSON output requirements, but their training samples contain different modality-specific information. M1 samples include maintenance-step text and associated document-image references, whereas M2 samples include video clips, step descriptions, and annotated key-frame references. During inference, M1 extracts events from maintenance text and document context, while M2 directly processes original videos to generate chronological event sequences. The same LoRA workflow was used to train both models. Details of the training datasets are provided in Section 4.1.

(3) Organization and validation of maintenance-event sequences

The structured events extracted by M1 and M2 are organized into separate process sequences because the text and video datasets describe different maintenance tasks. Events are first ordered by step numbers, video chronology, or explicit timestamps.

Within each task, the RapidFuzz Python library (version 3.14.6, Python 3.11.15) is used to identify candidate-entity matches based on string similarity. These candidates are further checked for MGO class compatibility to prevent incorrect matches between different entity types, such as tools and spare parts. Component identifiers, equipment context, and installation positions are examined to distinguish different entities of the same class. Referential expressions, such as "the component", are linked to preceding entities only

when their identities are clear. Candidate matches and referential links are then manually verified using MGO semantic relations, while ambiguous entity mentions remain separate.

MGO semantic relations are also used to verify prerequisite constraints and check the completeness of tools, spare parts and consumables, precautions, and other key elements. Incorrect or missing information is manually corrected or supplemented when it can be unambiguously verified against the original maintenance materials. These checks help reduce the propagation of identifiable extraction errors into the knowledge graph. Elements that remain missing or cannot be verified from the source materials are subsequently processed by the graph-based reasoning and completion stage.

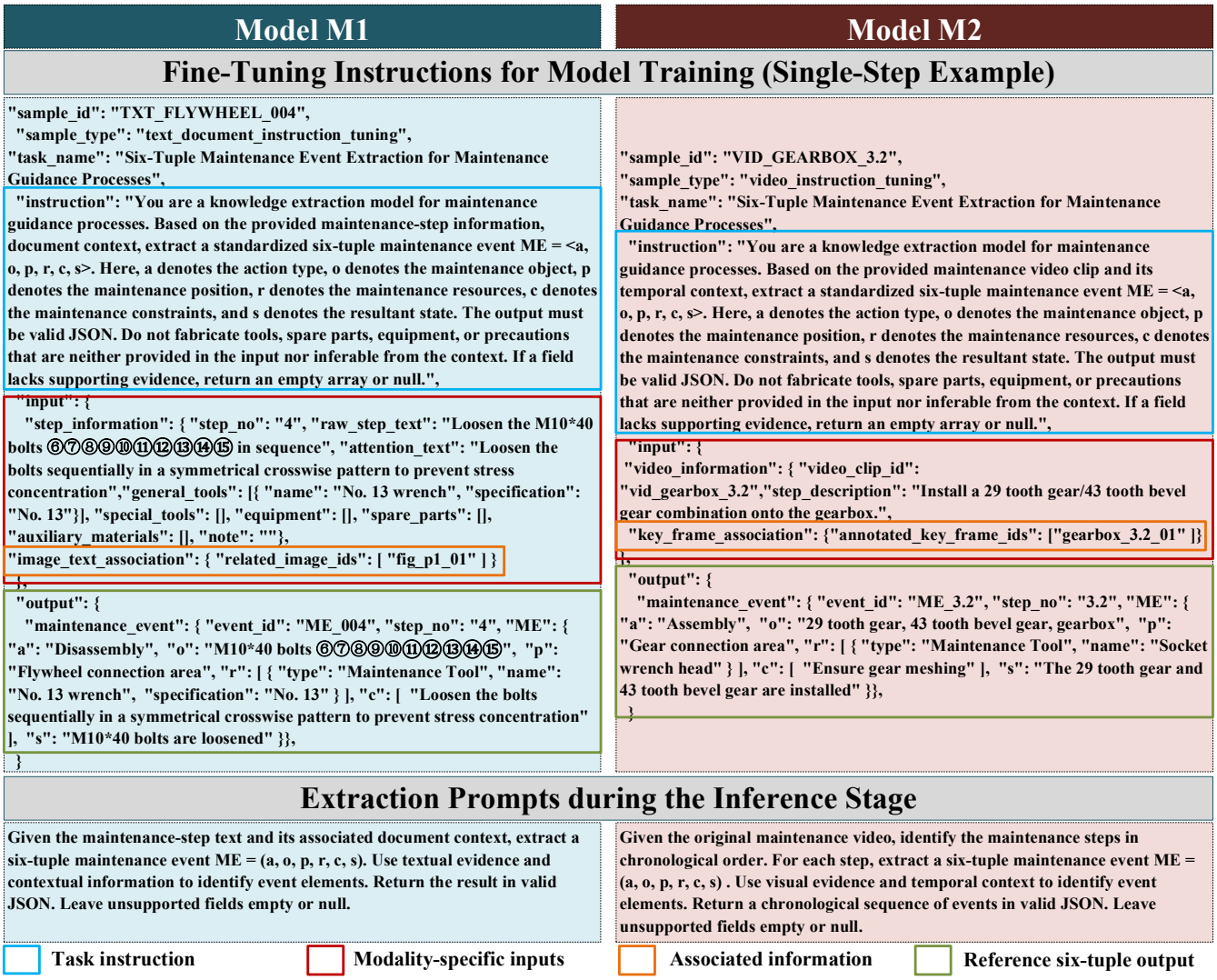


Figure 5. Representative LoRA fine-tuning samples and inference prompts for text (M1) and video (M2) maintenance-event extraction.

3.4. Graph-Reasoning-Based Completion of Missing Maintenance-Guidance Elements

The preceding method extracts standardized maintenance events from multimodal materials. However, incomplete source materials and extraction-model limitations leave some steps without essential tools, spare parts and consumables, or precautions. We therefore construct a maintenance-guidance-process knowledge graph and propose the MGO-STGCN. The model combines graph-node semantics, relation types, and step-order information to infer and complete missing maintenance elements.

3.4.1. Construction of the Maintenance-Guidance-Process Knowledge Graph

Compared with conventional databases or document-based storage, a knowledge graph explicitly represents multiple semantic associations among steps, objects, resources, and constraints. It supports path queries, rule validation, and graph-learning-based reasoning. In this study, the MGO constrains the schema layer. The text-derived six-tuple events extracted by M1 are mapped to nodes, edges, and attributes. Video-extracted events were not incorporated into the knowledge graph because the available video dataset was relatively small and several event elements were not directly observable, potentially introducing additional incompleteness into the graph.

Nodes represent maintenance steps, actions, objects, positions, tools, spare parts and consumables, precautions, and resultant states. Edges represent semantic associations between a step and its event elements, together with precedence relations between adjacent steps. The knowledge graph is formally represented as

$$G = (V, E, A) \tag{9}$$

where V, E and A denote the node, edge, and attribute sets, respectively. Each maintenance event is converted into entity–relation–entity triples using the relations defined by the MGO. Entity normalization and relation instantiation then populate the graph, which is stored in a Neo4j graph database (version 2025.03.0). This representation transforms maintenance-element completion into association prediction between a given step and candidate entities, providing structured input for graph reasoning.

3.4.2. Reasoning and Completion with MGO-STGCN

Node semantics, relation types, and temporal information jointly affect missing-element prediction in the maintenance-guidance-process knowledge graph. We therefore developed the MGO-STGCN architecture shown in Figure 6. Graph-feature representation and graph-convolutional encoding are followed by an MGO-constrained multi-head decoder. The three heads infer and complete tools, spare parts and consumables, and precautions, with precautions represented as prerequisite-constraint information.

(1) Graph-feature representation and encoding with semantic and temporal information

A BERT model first encodes the textual information of each graph node as a 768-dimensional semantic vector, producing the feature matrix

$$X = [x_1, x_2, \ldots x_n]^T \tag{10}$$

where x_i is the semantic feature of node i. Edge features are then constructed so that relation type and step order participate in graph convolution:

$$b_{ij} = [\text{onehot}(r_{ij}) \| \text{dir}_{ij} \| \Delta\text{pos}_{ij}] \tag{11}$$

Here, onehot(r_{ij}) is a one-hot encoding of the edge's relation type. The edge-direction feature $\text{dir}_{ij} \in \{0, -1, 1\}$ distinguishes the orientation between entities. The temporal distance between two maintenance-step nodes is denoted by $\Delta\text{pos}_{ij} \in N$. N denotes the set of nonnegative integers. When an edge connects adjacent steps, this value represents the step interval. When an edge connects a step to another entity, $\Delta\text{pos}_{ij} = 0$. The discrete relation features are mapped to a continuous edge weight ω_{ij}, from which the weighted adjacency matrix A_{ij}^{ω} is constructed:

$$\omega_{ij} = \text{sigmoid}\left(w_b^T b_{ij} + c_b\right) \tag{12}$$

$$A_{ij}^{\omega} = A_{ij} \cdot \omega_{ij} \tag{13}$$

where w_b and c_b are trainable parameters and A_{ij} is an element of the original adjacency matrix. Relation semantics and differences in step order are thus transformed into neighborhood-message weights, allowing the contribution of each relation type to the node representation to be learned. Self-loops are added to the weighted adjacency matrix, followed by symmetric degree normalization to obtain the normalized adjacency matrix used for graph convolution $\hat{A}$:

$$\tilde{A} = A^{\omega} + I, \quad \hat{A} = D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}} \tag{14}$$

where I is the identity matrix and D is the diagonal degree matrix of the self-loop weighted adjacency matrix. Neighborhood aggregation and feature transformation are then performed through three GCN layers:

$$H^{(l+1)} = \mathrm{ReLU}\left(\hat{A}H^{(l)}W^{(l)}\right), \quad H^{(0)} = X \tag{15}$$

where $W^{(l)}$ is the trainable weight matrix of layer l and $Z = H^{(3)}$ is the final node-embedding matrix. The weighted adjacency matrix allows a node to aggregate the semantic features of adjacent nodes while adjusting the strength of information propagation according to relation type and temporal distance. The resulting node representation therefore incorporates the maintenance-operation context.

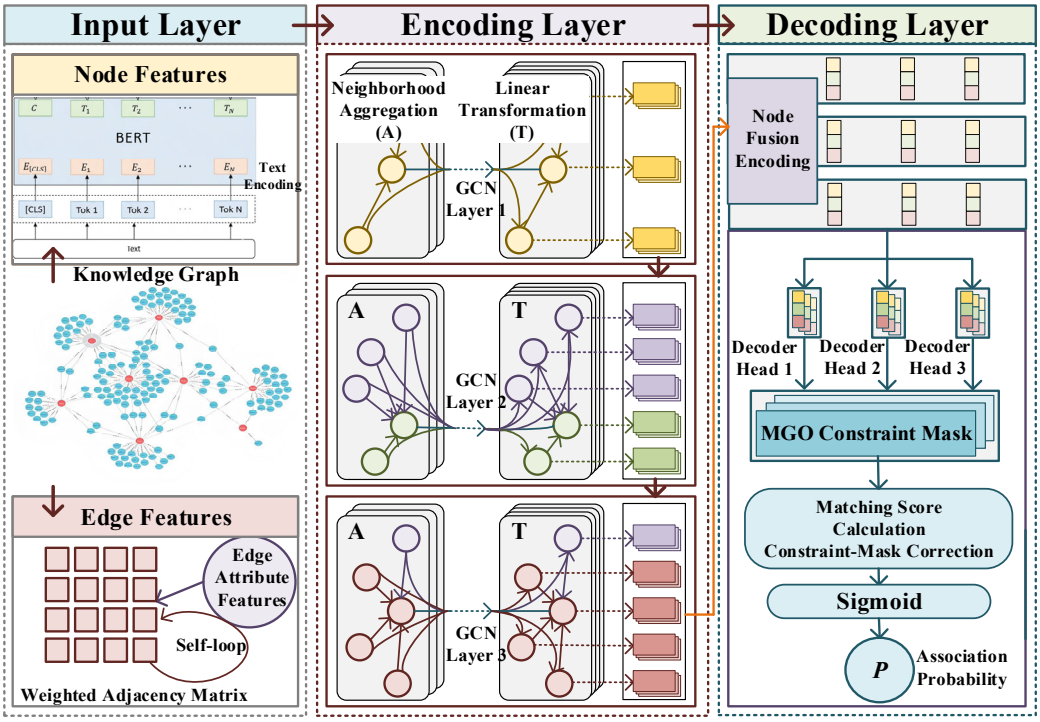


Figure 6. MGO-STGCN architecture for reasoning about and completing missing elements in a maintenance-guidance process.

(2) Multi-head entity prediction with MGO constraints

Tools, spare parts and consumables, and precautions have different semantic properties. Three independent decoding heads therefore calculate bilinear matching scores between each maintenance-step node and candidate entities of the corresponding type. The unified formulation is

$$s_{ij}^{r} = z_i^{T} W_r h_j^{r} + b_r \tag{16}$$

Here, $r \in \{1, 2, 3\}$ denotes the prediction type, corresponding to tools, spare parts and consumables, and precautions, respectively. z_i is the embedding of maintenance-step node i, and h_j^{r} is the embedding of candidate entity j for type r. The trainable parameters of the

corresponding decoding head are W_r and b_r. The bilinear mapping learns feature interactions between maintenance steps and candidate entities, thereby representing associations among maintenance actions, objects, resources, and constraints.

An MGO constraint mask $M_{ij}^r \in \{0, 1\}$ is introduced to suppress unreasonable predictions based only on statistical correlations. A Python-based rule-checking procedure is used to generate the mask from the MGO class hierarchy and applicable SWRL rules. The procedure retrieves maintenance-step and candidate-entity information from the Neo4j knowledge graph. It then checks their class memberships and semantic relations against the rule conditions. For example, the SWRL rule "CalibrationStep(?s) ∧ WeldingTool(?t) → InvalidCandidate(?s, ?t)" identifies welding tools as invalid candidates for calibration steps. When an explicit constraint violation is detected, the corresponding mask value is set to 0; otherwise, it is set to 1. The resulting binary mask is constructed before candidate-score correction. Equations (17) and (18) then calculate the corrected matching scores and association probabilities, respectively.

$$\tilde{s}_{ij}^r = s_{ij}^r + \lambda \log\left(M_{ij}^r + \varepsilon\right) \tag{17}$$

$$p_{ij}^r = \text{sigmoid}(\tilde{s}_{ij}^r) \tag{18}$$

where $\lambda = 0.3$ is the constraint strength and $\varepsilon = 1 \times 10^{-6}$ ensures numerical stability in the logarithm. This mechanism penalizes candidates that violate ontological constraints in the continuous score space. It reduces their predicted probabilities while retaining the neural network's ability to distinguish among valid candidates. Based on the MGO whose logical consistency was verified in Section 3.2, the constraint mask can mitigate the propagation of certain extraction errors by penalizing ontology-incompatible candidate predictions.

(3) Multitask loss optimization and missing-element completion

Completion of each entity type is a multilabel prediction task trained using binary cross-entropy. For prediction type r, the loss is

$$L_r = -\frac{1}{NC_r} \sum_{i=1}^{N} \sum_{j=1}^{C_r} [y_{ij}^r \log p_{ij}^r + (1 - y_{ij}^r) \log(1 - p_{ij}^r)] \tag{19}$$

where N is the number of maintenance-step nodes, C_r is the number of candidate entities of type r, the binary label y_{ij}^r is the ground-truth association, and p_{ij}^r is the predicted association probability defined in Equation (18). An MGO constraint loss further penalizes high probabilities assigned to invalid candidate entities, as defined in Equation (20).

$$L_{\text{MGO}} = \frac{\sum_{r=1}^{3} \sum_{i=1}^{N} \sum_{j=1}^{C_r} \left(1 - M_{ij}^r\right) p_{ij}^r}{\sum_{r=1}^{3} NC_r} \tag{20}$$

The model is jointly optimized using the following weighted multitask loss:

$$L = \alpha_1 L_{\text{tool}} + \alpha_2 L_{\text{mat}} + \alpha_3 L_{\text{note}} + \alpha_4 L_{\text{MGO}} \tag{21}$$

Here, α_1 to α_4 are the weights for the tool, spare parts and consumables, precaution, and MGO constraint losses, respectively. Each weight is set to 0.25. After training, candidate entities are selected using their association probabilities and type-specific thresholds. Thresholds of 0.65, 0.62, and 0.58 were selected on the validation set for tools, spare parts and consumables, and precautions, respectively. Predicted results are written back to the knowledge graph as triples, updating missing slots in the corresponding maintenance events. The completed event sequence is then associated with guidance presentation forms and formats. It can be converted into ARMG information comprising step descriptions, object prompts, resource information, and precautions.

4. Experimental Setup

Three groups of experiments evaluated the proposed method: maintenance-text event extraction, maintenance-video event extraction, and reasoning-based completion of maintenance-guidance processes. The experiments assessed extraction accuracy, process completeness, and computational efficiency.

4.1. Experimental Methods and Data

(1) Experiment 1: Maintenance-guidance-process extraction from text

Model M1 was trained on the 2813 maintenance-step samples from 100 tasks in Boeing 737 and Airbus A320 maintenance manuals, using an 8:2 training-to-validation split. The test set contained 232 steps from eight representative Airbus A320 maintenance-manual tasks and two diesel-engine component disassembly and assembly job cards. The maintenance manuals and maintenance job cards used in this experiment were obtained through a previous collaborative research project.

Comparison methods included BERT-BiLSTM-CRF variants and the general LLMs DeepSeek-V4-Flash and DeepSeek-V4-Pro. DeepSeek-V4-Pro was evaluated using both zero-shot and five-shot prompting. For the latter, five labeled maintenance-step samples were randomly selected from the training set as input–output demonstrations, without updating model parameters. Five trained annotators performed manual extraction, while M1 was run five times on the same three maintenance tasks to compare extraction efficiency.

(2) Experiment 2: Maintenance-guidance-process extraction from video

Model M2 was trained using 1479 annotated maintenance-step samples from 50 maintenance-task videos, with an 8:2 training-to-validation split. The test set comprised 78 steps from five third-person-view maintenance videos, including an engine flywheel-housing assembly and a Boeing 737 walk-around inspection. The video materials were obtained from two sources: laboratory recordings of flywheel-housing and transmission-gearbox assembly operations, and maintenance videos collected from online video resources. The externally sourced videos covered aviation, vehicle, and electrical-equipment maintenance tasks.

The comparison models were LLaVA-NeXT-Video-7B, InternVL2-8B, MiniCPM-V-2.6, and Qwen3-VL-8B. During testing, M2 and all comparison VLMs received the same original maintenance videos without prior step segmentation or manual annotations. Each model was instructed to extract maintenance events using the unified six-tuple representation. Three representative maintenance videos were also selected for a manual-extraction efficiency comparison.

(3) Experiment 3: Reasoning and completion of missing maintenance-guidance elements

A knowledge graph was constructed exclusively from the M1-extracted events corresponding to the 2813 text-based maintenance steps used in Experiment 1, without incorporating video-derived events. Tools, spare parts and consumables, and precautions were randomly masked in 20% of the steps. The resulting data were divided into training and validation sets at a ratio of 8:2. The test set (extracted by M1 from the test set in Experiment 1) contained manually verified complete maintenance events. Tools, spare parts and consumables, and precautions were separately masked in 50 steps for each element type. SWRL rule reasoning, TransE, RotatE, and R-GCN were used as comparison methods. Ablation experiments evaluated the contributions of the weighted adjacency matrix and the MGO constraint mask.

4.2. Experimental and Evaluation Parameters

(1) Experimental parameters

The experimental parameters are shown in Table 1.

Table 1. Experimental parameters.

Experiment	Model Category or Name	Model Parameters	Computing Environment
1	Sequence-labeling model	Maximum input length: 256; batch size: 32; learning rate: 2×10^{-5}; epochs: 20; weight decay: 0.01; BiLSTM hidden units: 256	AMD Ryzen 9 9950X3D; 48 GB RAM; NVIDIA GeForce RTX 5090 D, 32 GB
	General LLMs	Decoding temperature: 0; maximum generation length: 512	
	M1	Rank: 8; scaling factor: 16; dropout: 0.05; learning rate: 2×10^{-4}; epochs: 5	
2	General VLMs	Decoding temperature: 0; maximum generation length: 512	
	M2	Scaling factor: 32; learning rate: 1×10^{-4}; all other parameters as for M1	
3	TransE	Embedding dimension: 128; margin: 1.0; negative samples: 10; batch size: 1024; learning rate: 1×10^{-3}; epochs: 500; optimizer: Adam; L2 regularization: 1×10^{-5}	Intel Core Ultra 9 275HX; 32 GB RAM; NVIDIA GeForce RTX 5070 Ti, 12 GB
	RotatE	Margin: 6.0; all other parameters as for TransE	
	R-GCN	Embedding dimension: 128; hidden dimension: 128; GCN layers: 2; dropout: 0.2; batch size: 32; learning rate: 1×10^{-3}; epochs: 200; optimizer: AdamW; weight decay: 5×10^{-4}	
	MGO-STGCN	GCN layers: 3; selection thresholds: 0.65, 0.62, and 0.58; epochs: 140; all other parameters as for R-GCN	

(2) Evaluation metrics

Knowledge extraction performance was evaluated at both the slot and event levels. Slot-level evaluation used the F1 score for each of the six event slots and the macro-averaged F1 across all slots. Event-level evaluation used Event-F1 and the exact match rate (EMR). Event-F1 evaluates the correctness of the extracted event elements, whereas EMR measures the proportion of maintenance steps for which all six event slots are correctly extracted.

Text-extraction efficiency was evaluated using average extraction time (AET) per maintenance step. Video-extraction efficiency was evaluated using the real-time factor (RTF), defined as processing time divided by the original video duration. The processing time of M2 is measured from video frame sampling to the generation of the six-tuple maintenance-event sequence.

Graph-based completion was evaluated using precision (P), recall (R), F1, and average inference time per maintenance step for each of the three element types. Average inference time is defined as the total elapsed time from the start of program execution to the output of all completed elements, divided by the number of inferred steps. For MGO-STGCN, the inference time includes the computation and application of the MGO constraint mask. Precision, recall, and F1 are defined in Equations (22)–(24).

$$P = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{22}$$

$$R = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{23}$$

$$\mathrm{F1} = 2PR/(P + R) \tag{24}$$

Here, TP denotes a true positive, FP a false positive, FN a false negative, and TN a true negative.

5. Results and Discussion

5.1. Knowledge Extraction from Maintenance Text

5.1.1. Extraction Accuracy

Figure 7a,b compare the slot-level extraction results of M1 and the comparison methods. The traditional sequence-labeling models achieved macro-averaged F1 scores of 0.247–0.280. The zero-shot DeepSeek Flash and DeepSeek Pro methods achieved 0.383 and 0.433, respectively. Five-shot prompting improved the performance of DeepSeek-V4-Pro to 0.600, whereas M1 achieved 0.735, an improvement of 0.135 over the strongest comparison method. For the six slots, M1 achieved F1 scores of 0.87 for maintenance actions, 0.60 for maintenance resources, and 0.97 for prerequisite constraints, exceeding all comparison models. Traditional sequence-labeling models predict mainly from local text features and label-transition relations, which limit their ability to represent cross-sentence associations and compound semantics in maintenance steps. Few-shot prompting improved DeepSeek-V4-Pro, particularly for constraints (F1: 0.22 to 0.90), actions, and resources, by providing labeled examples of event slots. However, its limited gains for objects and positions suggest that in-context demonstrations alone cannot fully resolve domain-specific semantic ambiguities. LoRA allowed M1 to learn specialist terminology and the six-tuple event structure more comprehensively. Its F1 score for the maintenance-position slot was only 0.38, however, indicating that fine-grained localization within complex maintenance objects remains difficult.

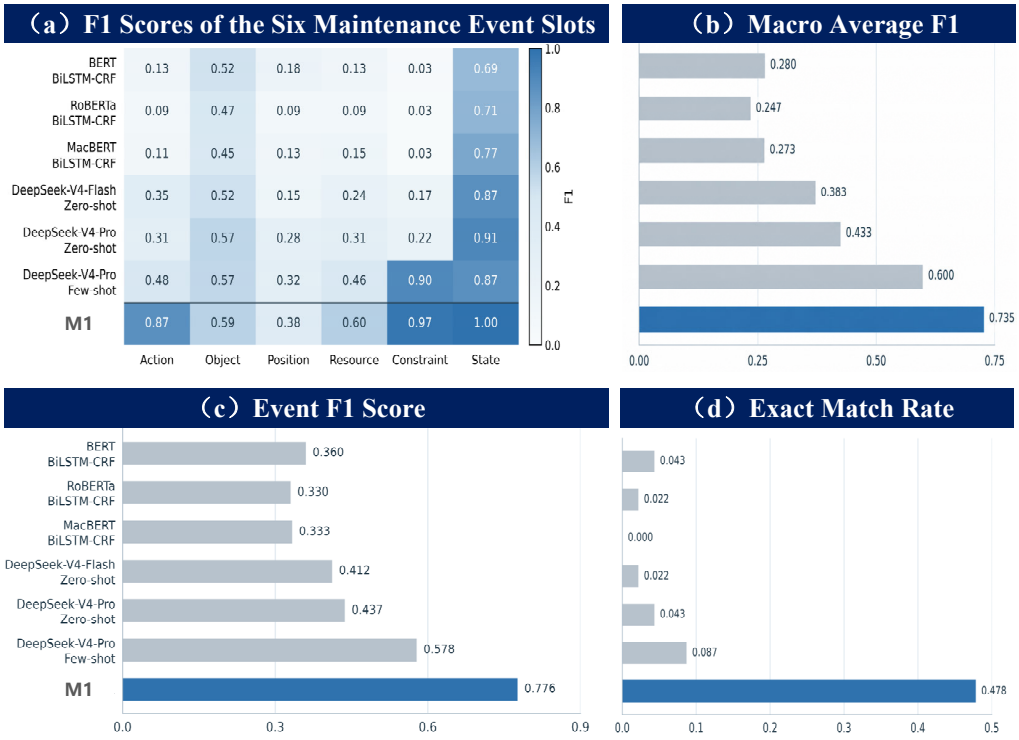


Figure 7. Accuracy comparison for maintenance-guidance knowledge extraction from text: (**a**,**b**) slot-level results and (**c**,**d**) event-level results. In (**b**–**d**), blue bars indicate the proposed M1 model, while gray bars indicate the comparison models.

Figure 7c,d present the event-level results. M1 achieved an Event-F1 of 0.776, exceeding the few-shot baseline by 0.198. Its EMR reached 0.478, substantially higher than the 0.087 achieved by the few-shot baseline. These results suggest that few-shot examples improve structured extraction by general LLMs, while domain-specific LoRA fine-tuning provides stronger overall usability of the extracted structured events.

5.1.2. Extraction Efficiency

Table 2 compares the extraction efficiency of M1 with manual annotation. M1 required 0.07–0.08 min per maintenance step, compared with 0.44–1.78 min for manual extraction. The results indicate that M1 substantially reduces the time required to convert maintenance text into structured events.

Table 2. Efficiency comparison for knowledge extraction from maintenance text.

Method	Maintenance Task	Steps	Extraction Time	AET	Mann–Whitney *p* Value
Manual extraction	Replace the fuel-tank filler-cap O-ring	14	22.95 ± 2.67	1.64	0.0079
M1			1.09 ± 0.08	0.08	
Manual extraction	Install the oil tank	31	55.10 ± 7.26	1.78	0.0079
M1			2.16 ± 0.21	0.07	
Manual extraction	Disassemble and inspect the electronic control box	9	3.93 ± 0.36	0.44	0.0079
M1			0.75 ± 0.05	0.08	

Note: Extraction time is reported as mean ± standard deviation (SD), in minutes.

5.2. Knowledge Extraction from Maintenance Videos

5.2.1. Extraction Accuracy

Figure 8a,b show the F1 score for each maintenance-event slot and the macro-average for all models. M2 achieved the best result for every slot and a macro-averaged F1 of 0.5471, which was 0.2252 higher than that of the general Qwen3-VL-8B model. Its F1 for prerequisite constraints was only 0.29. Safety conditions, procedural requirements, and resultant states cannot always be determined from visible actions, so the model still failed to recover every event element. Nevertheless, prerequisite constraints and resultant states were retained to ensure a complete evaluation of the six-tuple representation. Although some may be inferred from observable actions and procedural context, evaluating these challenging elements helps reveal the capabilities and limitations of video-based maintenance-event extraction.

Figure 8c,d show the event-level results. M2 achieved the highest Event-F1 and EMR among the comparison models, at 0.5764 and 0.2564, respectively. Together, the results indicate that extracting maintenance events from video requires not only local action or object recognition but also semantic associations among multiple event elements. Domain fine-tuning with key-frame object annotations and structured-output supervision improved performance on this task.

5.2.2. Extraction Efficiency

Table 3 shows that M2 achieved RTF values of 85.28%, 78.36%, and 82.00% on the three representative maintenance videos. Every value was below 100%, indicating that processing required less time than the original video duration. Manual extraction produced RTF values of 327.61%, 514.43%, and 426.00%, requiring several times the video duration for viewing, interpretation, and structured recording. At its current event-extraction accuracy, M2 therefore substantially reduces the time required to acquire knowledge from maintenance videos and supports near-real-time structured parsing and assisted guidance-process construction.

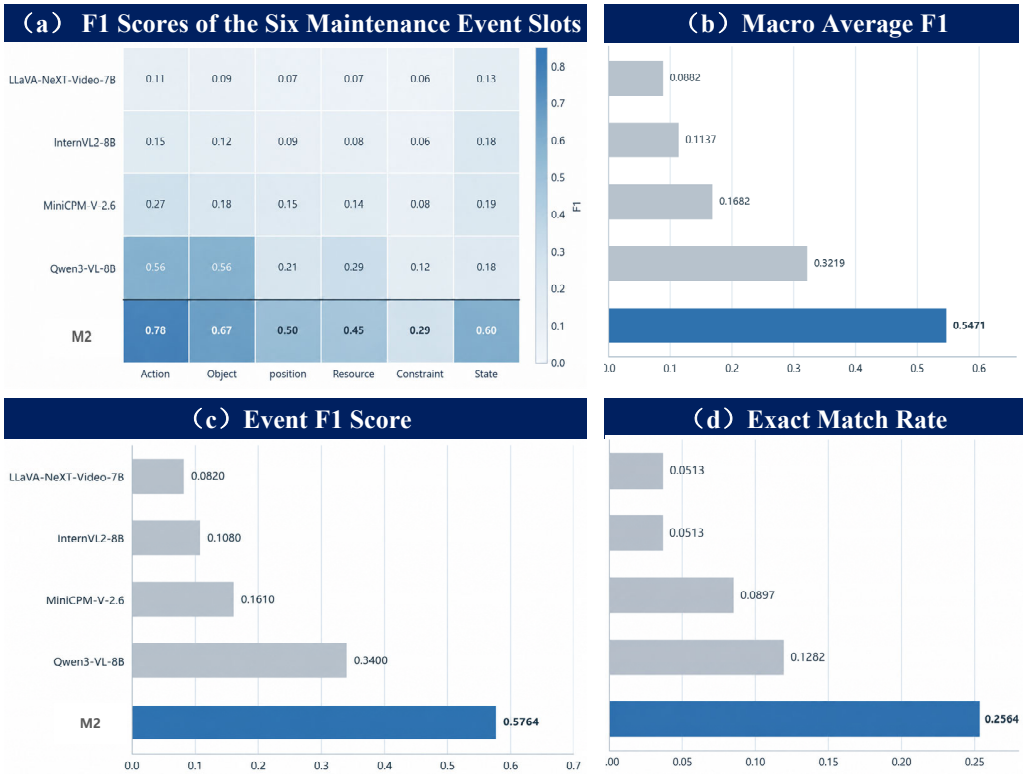


Figure 8. Accuracy comparison for maintenance-guidance knowledge extraction from video: (**a**,**b**) slot-level results and (**c**,**d**) event-level results. In (**b**–**d**), blue bars indicate the proposed M2 model, while gray bars indicate the comparison models.

Table 3. Efficiency comparison for knowledge extraction from maintenance videos.

Method	Maintenance Task	Video Duration	Extraction Time	RTF (%)	Mann–Whitney *p* Value
Manual extraction	Disassemble and inspect an engine flywheel housing	3.26	10.68 ± 4.13	327.61	0.0079
M2			2.78 ± 0.42	85.28	
Manual extraction	Assemble a turboshaft-engine transmission casing	3.05	15.69 ± 5.95	514.43	0.0079
M2			2.39 ± 0.38	78.36	
Manual extraction	Drain the A320 water system with electrical power on	0.50	2.13 ± 1.27	426.00	0.0079
M2			0.41 ± 0.14	82.00	

Note: Extraction time is reported as mean ± standard deviation (SD), in minutes.

5.3. Reasoning and Completion of Missing Maintenance-Guidance Elements

5.3.1. Ablation Analysis

To evaluate the contributions of relation semantics, temporal features, and ontological constraints, a standard GCN served as the baseline. The weighted adjacency matrix, the MGO constraint mask, and their combination were then added separately. As shown in Table 4, both components improved completion performance relative to the baseline, while their combination achieved the highest F1 scores across all three tasks. Compared with baseline model a, model b increased F1 by 0.080, 0.084, and 0.086 for tools, spare parts and consumables, and precautions, respectively. This indicates that weighted message passing based on relation semantics and step-order information improves inference by adjusting neighborhood contributions. Model c, which incorporated the MGO mask, increased F1 by 0.171, 0.167, and 0.157, respectively. Ontological constraints therefore improve completion accuracy by suppressing invalid candidates. The larger performance improvement achieved by the MGO constraint mask is attributed to its direct suppression of candidates that violate ontological constraints through the score correction mechanism in Equation (17). This

mechanism acts more directly on candidate predictions than the weighted adjacency matrix, which adjusts the contributions of neighboring nodes during aggregation. The complete model d achieved the best performance, with F1 scores of 0.799, 0.735, and 0.680 for the three tasks. These scores exceeded those of model a by 0.288, 0.277, and 0.254, respectively, demonstrating the complementary contributions of the weighted adjacency matrix and MGO constraint mask.

Table 4. Components and results of the ablation experiments.

Model	Baseline (GCN)	Weighted Adjacency	MGO Mask	Completed Element	Precision	Recall	F1
a	√	×	×	Tool	0.550	0.478	0.511
				Spare part or consumable	0.482	0.436	0.458
				Precaution	0.453	0.402	0.426
b	√	√	×	Tool	0.619	0.565	0.591
				Spare part or consumable	0.569	0.517	0.542
				Precaution	0.545	0.482	0.512
c	√	×	√	Tool	0.714	0.652	0.682
				Spare part or consumable	0.651	0.601	0.625
				Precaution	0.609	0.560	0.583
d	√	√	√	Tool	0.821	0.779	0.799
				Spare part or consumable	0.754	0.716	0.735
				Precaution	0.703	0.659	0.680

Note: "√" indicates that the corresponding component is included, whereas "×" indicates that it is excluded.

5.3.2. Completion Accuracy and Inference Efficiency

Figure 9 presents the training loss and validation F1 curves of the four completion models. TransE and RotatE showed relatively gradual improvements in validation F1, whereas R-GCN achieved better completion performance through relation-aware neighborhood aggregation. MGO-STGCN attained a higher validation F1 within fewer training epochs and exhibited a relatively smooth loss trajectory. These observations suggest that the combined semantic–temporal representation and ontology-constrained optimization may contribute to its training performance under the evaluated settings.

Table 5 compares the completion methods. MGO-STGCN achieved the highest precision, recall, and F1 across all three element types while maintaining the lowest average inference time. SWRL rule reasoning was relatively fast but had the lowest completion accuracy. TransE and RotatE improved performance by representing entities and relations in vector spaces. R-GCN used relation-aware neighborhood aggregation and achieved F1 scores of 0.637, 0.583, and 0.552 for tools, spare parts and consumables, and precautions, respectively. MGO-STGCN achieved F1 scores of 0.799, 0.735, and 0.680 for the corresponding tasks, exceeding R-GCN by 0.162, 0.152, and 0.128, respectively. Its average inference time was 0.130 s per step, the shortest among the evaluated methods.

The efficiency advantage of MGO-STGCN is partly attributed to its relation–temporal edge weighting mechanism. Each edge feature combines the MGO relation type, relation direction, and step-order distance. These features are then mapped into a weighted adjacency matrix, allowing relation semantics, direction, and temporal information to jointly influence neighborhood aggregation. Compared with R-GCN, which employs relation-specific transformation matrices, MGO-STGCN reduces relation-specific parameters and avoids repeated relation-wise feature transformations, thereby improving inference efficiency. Nevertheless, differences in network architecture and the implementation of neighborhood aggregation may also affect inference speed.

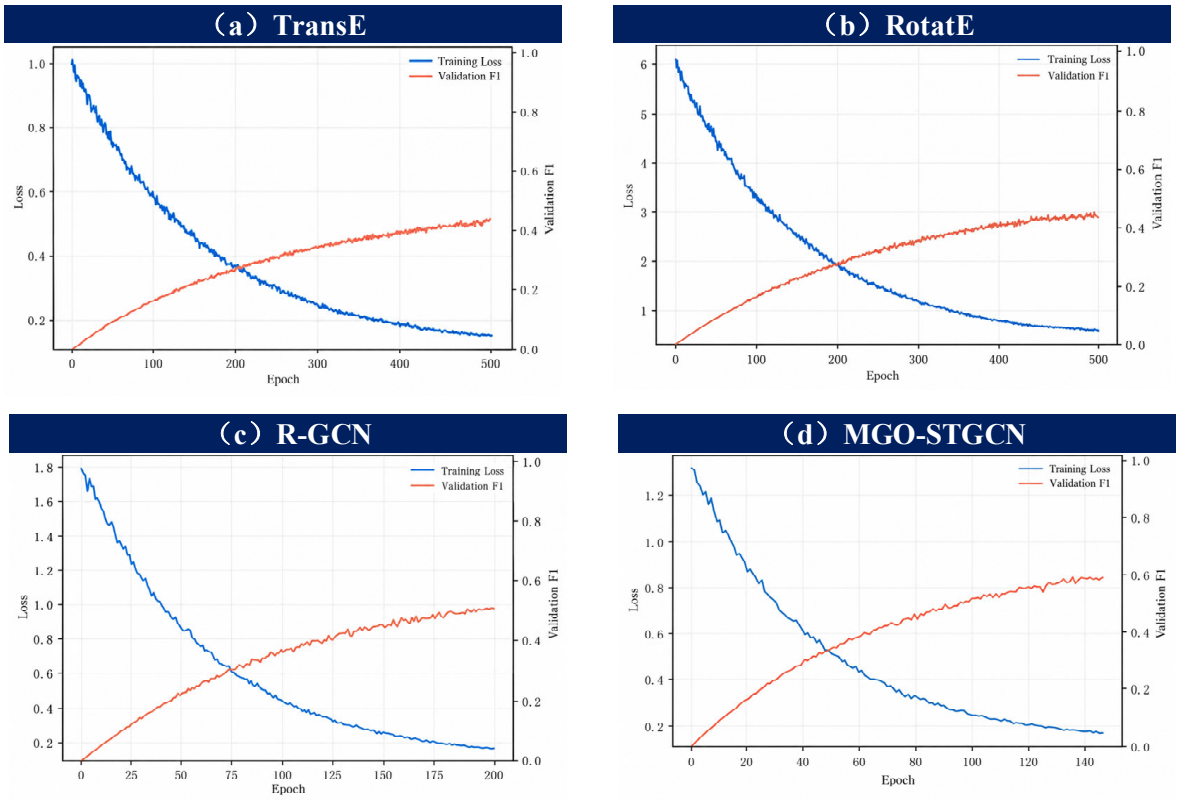


Figure 9. Training loss and validation F1 curves of the reasoning and completion models.

Table 5. Performance and inference efficiency of knowledge-graph reasoning and completion methods.

Method	Completed Element	Precision	Recall	F1	Average Inference Time (s/step)
SWRL rule reasoning	Tool	0.456	0.391	0.421	0.361
	Spare part or consumable	0.390	0.362	0.375	
	Precaution	0.379	0.317	0.345	
TransE	Tool	0.570	0.529	0.549	5.216
	Spare part or consumable	0.545	0.481	0.511	
	Precaution	0.501	0.449	0.474	
RotatE	Tool	0.591	0.565	0.578	9.089
	Spare part or consumable	0.558	0.527	0.542	
	Precaution	0.515	0.480	0.497	
R-GCN	Tool	0.669	0.608	0.637	3.230
	Spare part or consumable	0.607	0.560	0.583	
	Precaution	0.589	0.519	0.552	
MGO-STGCN	Tool	0.821	0.779	0.799	0.130
	Spare part or consumable	0.754	0.716	0.735	
	Precaution	0.703	0.659	0.680	

6. Conclusions

This study developed a formal modeling framework that integrates the Maintenance Guidance Ontology, VLMs fine-tuned on maintenance-domain knowledge, and MGO-STGCN to extract, represent, and complete maintenance-guidance knowledge from multi-modal materials. The framework organizes maintenance actions, resources, and constraints into a unified event representation and uses knowledge-graph-based association prediction to infer missing elements in maintenance steps.

On the current test sets, text-extraction model M1 achieved a slot-level macro-averaged F1 of 0.735 and an Event-F1 of 0.776. Video-extraction model M2 achieved a macro-averaged F1 of 0.5471 and an Event-F1 of 0.576. M1 required no more than 0.08 min per step, while the RTF of M2 was below 1 for all three representative videos. MGO-STGCN achieved F1 scores of 0.799, 0.735 and 0.680 for tools, spare parts and consumables, and precautions, respectively. Its average inference time was 0.130 s per step. The ablation experiments further showed that relation–temporal weighted propagation and MGO semantic constraints have complementary effects on candidate-entity association prediction.

The current models remain limited by the number of annotated samples and the completeness of the source materials. Prerequisite constraints and resultant states that are not directly observable in maintenance videos still reduce event completeness. The reasoning model was evaluated only on three missing-element types and has not yet been extended to missing-step prediction. Future work will explore MGO-guided cross-modal step alignment and event fusion using paired text–video data to address differences in step granularity. It will also investigate methods for acquiring constraints and state information that cannot be directly observed in video. Knowledge-graph completion will be extended to missing steps and other process relations. These developments will further evaluate the applicability of formal guidance processes in operational AR maintenance.

Author Contributions: Conceptualization, H.W.; Methodology, H.W., Y.Y. and X.W.; Validation, H.W., Y.Y. and Z.G.; Investigation, H.W.; Resources, D.Z.; Data curation, Z.G. and Y.Y.; Writing—original draft, H.W.; Writing—review and editing, H.W. and Z.G.; Visualization, X.W. and Y.Y.; Supervision, X.W. and Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the State Key Laboratory of Reliability and Environmental Engineering Technology under Grant Number 12700002025106001.

Data Availability Statement: The maintenance documents and externally sourced videos used in this study are subject to third-party copyright and redistribution restrictions and cannot be publicly shared by the authors. Access to these materials requires permission from the respective rights holders. The two laboratory-recorded videos of flywheel-housing assembly and transmission-gearbox assembly are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the domain experts and annotators for their valuable assistance in preparing and validating the maintenance datasets used in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Competency questions (CQs) for evaluating the MGO.

CQ	MGO Knowledge Layer	Competency Question	Relevant MGO Relations
CQ1	Fault knowledge	Which failure modes are associated with a specified failure location?	HasFailureMode
CQ2		Which failure causes, parent system, and system functions are associated with a given failure location?	LeadTo; PartOf; Realize
CQ3	Maintenance process	Which maintenance procedures are associated with eliminating a specified failure mode?	Eliminate
CQ4		Which standardized maintenance-step categories are defined in the MGO?	SubClassOf hierarchy

Table A1. *Cont.*

CQ	MGO Knowledge Layer	Competency Question	Relevant MGO Relations
CQ5	Resources and personnel	Which tools, equipment, spare parts, and consumables are required by a specified maintenance procedure?	Need; resource hierarchy
CQ6		Which personnel execute a specified maintenance procedure, and which maintenance criteria do they follow?	Execute; AbideBy
CQ7		Can tool-use relations be inferred from the procedures executed by personnel and the corresponding resource requirements?	Use
CQ8	Guidance information	What guidance information and instructional content are associated with a specified maintenance step?	HasGuidanceInformation; GuidanceContent
CQ9		Which presentation form can be inferred for a guidance-information item from its identifier?	PresentedAs
CQ10		Which information format and display parameters can be inferred for a guidance-information item from its identifier?	HasFormat

References

1. Xue, Z.; Yang, J.; Chen, R.; He, Q.; Li, Q.; Mei, X. AR-assisted guidance for assembly and maintenance of avionics equipment. *Appl. Sci.* **2024**, *14*, 1137. [CrossRef]
2. Zhao, X.; Liu, K.; Fu, H.; Wang, Y. Review of augmented reality maintenance and registration in aerospace industry. *J. Manuf. Process.* **2025**, *151*, 1018–1041. [CrossRef]
3. Hu, J.Y.; Zhao, G.; Xiao, W.L.; Li, R. AR-based deep learning for real-time inspection of cable brackets in aircraft. *Robot. Comput.-Integr. Manuf.* **2023**, *83*, 102574. [CrossRef]
4. Liang, Y.C.; Shiu, S.C.; Liu, C.W. Enhancing yield and process efficiency through dual Internet of Things and augmented reality for AI-driven human-machine interaction in centering mass production. *J. Manuf. Syst.* **2025**, *82*, 26–41. [CrossRef]
5. Bahari, M.A.; Salleh, S.; Kiram, M.K.A. Augmented reality assisted maintenance and monitoring at on-premise data center. In Proceedings of the 9th International Conference on Information Technology Convergence and Services (ITCSS 2023), Copenhagen, Denmark, 28–29 January 2023; pp. 49–58.
6. Nagy, A.; Spyridis, Y.; Mills, G.J.; Argyriou, V. User experience evaluation of AR-assisted industrial maintenance and support applications. In Proceedings of the 2024 10th International Conference on Virtual Reality (ICVR), Bournemouth, UK, 24–26 July 2024; pp. 270–277.
7. Gómez-Cambronero, Á.; Miralles, I.; Tonda, A.; Remolar, I. Immersive virtual-reality system for aircraft maintenance education: A case study. *Appl. Sci.* **2023**, *13*, 5043. [CrossRef]
8. Zhou, Q.; Zhou, D.; Wang, Y.; Guo, Z.; Dai, C. Knowledge reuse for ontology modelling and application of maintenance motion state sequence. *J. Ind. Inf. Integr.* **2024**, *41*, 100659. [CrossRef]
9. Shim, M.; Choi, H.; Koo, H.; Um, K.; Lee, K.-H.; Lee, S. OmEGa (Ω): Ontology-Based Information Extraction Framework for Constructing Task-Centric Knowledge Graph from Manufacturing Documents with Large Language Model. *Adv. Eng. Inform.* **2025**, *64*, 103001. [CrossRef]
10. Gao, Y.; Xiong, G.; Li, H.; Richards, J. Exploring bridge maintenance knowledge graph by leveraging GrapshSAGE and text encoding. *Autom. Constr.* **2024**, *166*, 105634. [CrossRef]
11. Zhou, Q. Research on Virtual Maintenance Parameterized Design Method Based on Knowledge Engineering. Ph.D. Thesis, Beihang University, Beijing, China, 2025; pp. 4–5.
12. Lin, R.; Wang, H.; Wang, J.; Wang, N. Knowledge representation and reuse model of civil aircraft structural maintenance cases. *Expert Syst. Appl.* **2023**, *216*, 119460. [CrossRef]

13. Chen, Y.; Liu, K.; Wang, S. Fault knowledge graph construction for aviation equipment based on BiGRU-Attention improvement. *Acta Aeronaut. Astronaut. Sin.* **2024**, *45*, 229916.
14. Huang, X.; Yang, C.; Zhang, Y.; Lou, S.; Kong, L.; Zhou, H. Ontology Guided Multi-Level Knowledge Graph Construction and Its Applications in Blast Furnace Ironmaking Process. *Adv. Eng. Inform.* **2024**, *62*, 102927. [CrossRef]
15. Feng, L.; Fang, Q.; Hu, Y.; Gu, C.; Liu, J. Ontology Construction of Knowledge Graph for Early Warning Equipment Maintenance and Support Field. In Proceedings of the 2024 IEEE 2nd International Conference on Sensors, Electronics and Computer Engineering (ICSECE), Jinzhou, China, 29–31 August 2024; pp. 587–591.
16. Kong, X.; Li, Y.; Fan, M.; Shi, J.; Wei, L.; Qu, S. Automated Knowledge Mining and Knowledge Graph Reasoning for Aircraft Engine Maintenance. In Proceedings of the 2024 6th International Conference on Pattern Recognition and Intelligent Systems (PRIS 2024), Hong Kong, China, 25–27 July 2024; pp. 35–40.
17. Xu, Y.; Peng, T.; Zhong, R.Y.; Lim, K. An Intelligent Fault Diagnosis Approach for Construction Machinery Hydraulic Systems Based on Knowledge Graph. *Adv. Eng. Inform.* **2026**, *71*, 104360. [CrossRef]
18. Geng, H.; Qing, H.; Hu, J.; Huang, W.; Kang, H. A named entity recognition method for Chinese vehicle fault repair cases based on a combined model. *Electronics* **2025**, *14*, 1361. [CrossRef]
19. Wei, Z.; Qu, S.; Zhao, L.; Shi, Q.; Zhang, C. A position- and similarity-aware named entity recognition model for power equipment maintenance work orders. *Sensors* **2025**, *25*, 2062. [CrossRef]
20. Dong, J.; Li, J.; Li, Y.; Zhang, W.; Zhang, Z.; Guo, C.; Dang, Y.; Chen, M.; Du, J. Knowledge extraction and alignment for mine ventilation: A knowledge graph construction framework based on large language models. *Eng. Appl. Artif. Intell.* **2025**, *162*, 112726. [CrossRef]
21. Giordano, V.; Fantoni, G. Decomposing maintenance actions into sub-tasks using natural language processing: A case study in an Italian automotive company. *Comput. Ind.* **2025**, *164*, 104186. [CrossRef]
22. Li, X.; Li, G.; Wang, T.; Dong, Z.; Qin, Z.; Chu, F. Knowledge extraction and retrieval-augmented generation for intelligent maintenance of wind power equipment based on graph attention networks. *Chin. J. Mech. Eng.* **2025**, *39*, 100141. [CrossRef]
23. Vico, L.; Polito, M.; Duarte, L.; Pastorelli, S.; Gastaldi, L.; Neto, P. Online classification of human gestures through event camera data using a 3DCNN. In Proceedings of the 2025 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Funchal, Portugal, 2–3 April 2025; pp. 52–57.
24. Zong, X.; Dong, S.; Guo, J. MSCDGCN: Multi-scale central difference graph convolution network for skeleton-based action recognition. *J. King Saud Univ. Comput. Inf. Sci.* **2025**, *37*, 313. [CrossRef]
25. Zhang, X.; Tan, C.; Yuan, Y.; Yan, J. Hierarchical intertwined graph representation learning for skeleton-based action recognition. *Sci. Rep.* **2025**, *15*, 35447. [CrossRef]
26. Deng, W.; He, S.; Wu, X.; Wan, Y.; Xiang, C.; Yao, Z.; Mi, H. AI Extraction of Disassembly Videos for Generating AR Repair Instructions. In Proceedings of the Adjunct Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct 2025), Busan, Republic of Korea, 28 September–1 October 2025; Article 72, pp. 1–3.
27. Ren, S.; Yao, L.; Li, S.; Sun, X.; Hou, L. TimeChat: A Time-Sensitive Multimodal Large Language Model for Long Video Understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 14313–14323.
28. Guillemin, S.; Roxin, A.; Dujourdy, L.; Journaux, L. Weighted Horn Clause: Extending SWRL to Model Antecedents' Importance and Handle Missing Data. *Expert Syst. Appl.* **2025**, *293*, 128464. [CrossRef]
29. Yang, X.; Li, H.; Xu, Y.; Deng, Y.; Bai, Y.; Liu, S. Hybrid semantic-structural graph neural network for fault knowledge graph completion in railway operational equipment. *Eng. Appl. Artif. Intell.* **2025**, *162*, 112490. [CrossRef]
30. Xia, L.; Liang, Y.; Leng, J.; Zheng, P. Maintenance Planning Recommendation of Complex Industrial Equipment Based on Knowledge Graph and Graph Neural Network. *Reliab. Eng. Syst. Saf.* **2023**, *232*, 109068. [CrossRef]
31. Otte, J.; Beverley, J.; Ruttenberg, A. Basic Formal Ontology. *Appl. Ontol.* **2022**, *17*, 17–43. [CrossRef]
32. Jensen, M.; De Colle, G.; Kindya, S.; More, C.; Cox, A.P.; Beverley, J. The Common Core Ontologies. In Formal Ontology in Information Systems: Proceedings of the 14th International Conference (FOIS 2024), Enschede, The Netherlands, 15–19 July 2024; pp. 47–58. [CrossRef]
33. Kulvatunyou, B.; Wallace, E.; Kiritsis, D.; Smith, B.; Will, C. The Industrial Ontologies Foundry Proof-of-Concept Project. In *Advances in Production Management Systems: Smart Manufacturing for Industry 4.0*; Moon, I., Lee, G.M., Park, J., Kiritsis, D., von Cieminski, G., Eds.; Springer: Cham, Switzerland, 2018; Volume 536, pp. 402–409. [CrossRef]
34. Industrial Ontologies Foundry. Maintenance Reference Ontology. Available online: https://spec.industrialontologies.org/portal/release/202301/maintenance/MaintenanceReferenceOntologyOntology.html (accessed on 15 October 2026).

35. Wang, X. Enhancing visual communication design and customization through the CLIP contrastive language-image model. *Int. J. Adv. Comput. Sci. Appl.* **2025**, *16*, 440–449. [CrossRef]
36. Tan, J.; Wang, Y.; Wu, G.; Wang, L. Temporal Perceiver: A general architecture for arbitrary boundary detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12506–12520. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.