

Article

KG-APC: Knowledge Graph-Guided Adaptive Prototype Correction for Few-Shot Entity Recognition in Industrial Maintenance Information Systems

Peng Du , Xiaoying Gao and Yang Xiang *

School of Computer Science and Technology, Tongji University, Shanghai 201804, China;
pengdu94@tongji.edu.cn (P.D.); gxying@tongji.edu.cn (X.G.)

* Correspondence: shxiangyang@tongji.edu.cn

Abstract

Industrial maintenance and fault-diagnosis systems generate textual records, such as maintenance work orders, service requests, causal analyses, and troubleshooting solutions. These records contain domain-specific named entities that provide valuable knowledge for intelligent monitoring, fault diagnosis, maintenance decision support, and industrial knowledge graph construction. However, in practical industrial environments, maintenance records are strongly associated with specific equipment types, production processes, fault modes, and enterprise-specific terminology. As a result, entity schemas vary across systems, new entity types emerge with equipment updates, and high-quality annotation requires substantial domain expertise. These factors make it difficult to obtain sufficient labeled samples for each industrial entity type. Under such low-resource conditions, conventional supervised named entity recognition (NER) models tend to suffer from unstable entity boundary detection and biased entity representations. To address these challenges, this paper proposes a boundary-aware knowledge graph-guided adaptive prototype correction framework for few-shot NER in industrial maintenance information systems. The proposed framework first introduces a boundary-aware span detection mechanism to improve entity localization in noisy and irregular maintenance texts. A knowledge graph-guided adaptive prototype correction module is then designed to construct entity class prototypes from limited support examples, reducing prototype bias caused by sparse annotations. Experiments are conducted on two representative industrial datasets, MaintIE and CFDK, covering maintenance short texts and fault-diagnosis records. Experimental results show that the proposed framework achieves an average Micro-F1 improvement of 1.89 percentage points over the strongest compared baseline across 12 episodic settings on the two industrial datasets: three MaintIE coarse-grained settings, six MaintIE fine-grained settings, and three CFDK settings. The ablation and sensitivity analyses further indicate that boundary-aware span modeling and KG-guided prototype correction jointly contribute to low-resource entity classification. This study provides a data-efficient information extraction solution for AI-enabled industrial knowledge acquisition, fault diagnosis, and maintenance decision support.



Academic Editor: Arkaitz Zubiaga

Received: 18 June 2026

Revised: 17 July 2026

Accepted: 17 July 2026

Published: 24 July 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.

Keywords: few-shot named entity recognition; industrial artificial intelligence; expert system; knowledge graph

1. Introduction

Artificial intelligence is increasingly becoming a core enabling technology for industrial digitalization. In industrial operation and maintenance scenarios, AI techniques can automatically mine valuable information from maintenance records, fault reports, service requests, and troubleshooting logs, thereby reducing dependence on manual analysis and improving the efficiency of knowledge acquisition. As an important technical foundation for industrial AI, knowledge graphs provide a structured semantic network for representing entities, attributes, and their relationships in graph form [1,2]. Large-scale open knowledge graphs such as Wikidata further organize entity names, aliases, descriptions, categories, and semantic relations, making them useful sources of external knowledge for label semantic enrichment [3]. Since industrial knowledge acquisition usually starts from domain-specific named entities, such as equipment, components, fault phenomena, fault causes, and repair actions, named entity recognition for industrial maintenance texts has become a fundamental task in AI-enabled industrial information systems [4,5].

Recent industrial large language model research has extended foundation-model adaptation to mechanical fault diagnosis under variable operating conditions. In particular, Xu et al. proposed VOC-LLM, which combines multidimensional vibration-feature fusion, variable-speed encoding, lightweight domain adaptation, and fault-semantic alignment to identify fault types and quantify fault severity under changing operating conditions [6]. This signal-oriented diagnostic direction is complementary to the textual information extraction studied here: industrial maintenance systems also contain work orders, service requests, causal descriptions, and troubleshooting records whose equipment, fault, cause, action, and property entities must be converted into structured knowledge. This complementary requirement motivates a lightweight and reproducible few-shot NER framework for low-resource maintenance texts.

Named entity recognition (NER) is a fundamental task for transforming unstructured text into structured knowledge. In industrial maintenance and fault-diagnosis scenarios, NER aims to identify domain-specific entities such as equipment names, components, fault phenomena, fault causes, service requests, repair actions, operational states, and physical properties. Accurate entity recognition provides the basis for downstream relation extraction, event extraction, fault knowledge graph construction, case retrieval, and intelligent maintenance recommendation. Therefore, industrial NER is not merely a conventional natural language processing task but an essential component of AI-driven knowledge acquisition and decision support in engineering applications [7,8].

Despite its practical importance, NER for industrial maintenance and fault-diagnosis records remains difficult. First, industrial textual records are usually written by field engineers or operators in concise and irregular forms. They often contain abbreviations, domain-specific terminology, incomplete grammar, inconsistent expressions, and noisy descriptions. Second, entity boundaries are frequently ambiguous. A fault phenomenon or troubleshooting solution may be expressed as a long phrase, while equipment names and component names may overlap or appear in nested structures. Third, high-quality annotation is expensive because it requires domain expertise. Unlike open-domain NER datasets, industrial datasets are typically small, proprietary, and highly dependent on specific equipment types and maintenance procedures. These factors make it difficult for fully supervised NER models to achieve stable performance across different industrial scenarios [4,5].

Existing industrial NER studies have mainly relied on traditional sequence labeling architectures, such as CRF-based models, BiLSTM-CRF models, and pretrained language model-based variants. These methods have achieved promising results when sufficient labeled data are available. However, their performance often degrades under low-resource

conditions because they require abundant annotated examples to learn reliable entity boundaries and type-specific semantic representations. In addition, standard sequence labeling methods usually couple boundary detection and entity type classification into a unified token-label prediction process. This design may amplify boundary errors in noisy industrial texts and restrict the model's ability to generalize to unseen or low-frequency entity types [5,9–16].

Few-shot NER provides a promising direction for industrial information extraction because it aims to recognize new entity types using only a few labeled examples. Metric-based and prototype-based few-shot learning methods are particularly attractive, as they classify query entities by comparing them with class prototypes constructed from support examples. However, applying few-shot NER directly to industrial maintenance and fault-diagnosis records still faces several limitations. First, limited support examples may produce biased or unstable prototypes, especially when entity mentions are semantically diverse within the same type. Second, many few-shot methods focus primarily on entity type classification and pay insufficient attention to boundary uncertainty, which is a major error source in industrial texts. Third, domain knowledge, such as label semantics, equipment-fault associations, and industrial conceptual structures, is often underutilized, although such knowledge can provide useful priors for entity discrimination in low-resource settings [17–27]. Recent knowledge-enhanced few-shot NER methods, such as KCL [28] and KGen [29], have shown that knowledge graphs can improve representation discrimination and label semantic modeling. However, industrial few-shot NER still requires a mechanism that uses knowledge not only to enrich representations, but also to correct unstable prototypes after reliable span boundaries have been obtained.

To address these problems, this paper proposes a boundary-aware knowledge graph-guided adaptive prototype correction framework for few-shot NER in industrial maintenance and fault-diagnosis textual records. The central idea is to decouple entity localization from entity type discrimination while enhancing both stages with low-resource representation learning. Specifically, the model first employs a boundary-aware span detection mechanism to identify candidate entity spans from noisy industrial text. This mechanism is designed to reduce propagation of token-level boundary errors and improve the model's sensitivity to entity boundary cues. Then, a knowledge graph-guided adaptive prototype correction module constructs class representations from support instances and adaptively refines them according to query context and domain semantic priors. By introducing knowledge-guided prototype adjustment, the framework can better capture the semantic characteristics of industrial entity types even when only a few labeled examples are available.

The proposed framework is evaluated on two representative industrial datasets, MaintIE and CFDK, as well as two widely used benchmark datasets for few-shot named entity recognition, Few-NERD and CoNLL-2003. MaintIE focuses on information extraction from maintenance short texts and provides fine-grained annotations for industrial maintenance entities. CFDK focuses on knowledge extraction from compressor maintenance log sheets and is closely related to equipment fault diagnosis. These two industrial datasets cover different textual forms, entity schemas, and application contexts, making them suitable for evaluating the robustness and generalization ability of few-shot industrial NER methods. In addition, Few-NERD and CoNLL-2003 are used to further verify the effectiveness of the proposed method under general few-shot NER settings [4,5,30,31]. In the experiments, multiple N-way K-shot settings represent realistic low-resource scenarios. The proposed model is compared not only with text-based and span-based few-shot NER baselines but also with knowledge-enhanced few-shot NER baselines, including KCL and KGen, so that the

contribution of adaptive prototype correction can be evaluated against methods that also exploit KG information.

The main contributions of this paper are summarized as follows.

- We propose a boundary-aware knowledge graph-guided adaptive prototype correction framework for few-shot NER. The proposed framework explicitly addresses two key challenges in low-resource entity recognition, namely, boundary detection errors and prototype instability under scarce supervision. By integrating boundary-aware span modeling with knowledge-guided prototype correction, the framework improves both entity localization and entity type discrimination in few-shot scenarios.
- We design a boundary-aware span detector that separates class-independent entity localization from entity type classification. Instead of directly assigning entity types to all tokens, the detector first identifies candidate entity spans and filters out irrelevant non-entity tokens. This design reduces the interference caused by noisy contextual words and provides higher-quality candidate spans for subsequent metric-based classification.
- We introduce a dynamic prototype correction mechanism with a K-aware gate. This mechanism adaptively balances support-set evidence and structured semantic prior information derived from the knowledge graph. When labeled support examples are extremely limited, the model can rely more on semantic priors to alleviate prototype bias; when more support examples are available, it can assign greater importance to data-driven prototype representations.
- We provide a comprehensive experimental design to evaluate the effectiveness and robustness of the proposed method. Experiments are conducted on standard few-shot NER benchmarks and industrial maintenance-related datasets under multiple N-way K-shot settings. In addition, ablation studies and K-shot sensitivity analysis are performed to verify the contribution of each module and to examine the model's behavior under different levels of supervision, especially in low-resource industrial maintenance information extraction scenarios.

The remainder of this paper is organized as follows. Section 2 reviews related work on industrial information extraction, named entity recognition, and few-shot learning. Section 3 introduces the proposed boundary-aware knowledge graph-guided adaptive prototype correction framework. Section 4 describes the experimental settings, datasets, baselines, and evaluation metrics. Section 5 presents the experimental results and analysis. Section 6 concludes the paper and discusses future research directions.

2. Related Work

2.1. Few-Shot NER and Metric Learning

Few-shot named entity recognition aims to recognize new entity types with only a small number of labeled examples. Compared with fully supervised NER, it is more suitable for domain-specific scenarios where annotation is expensive and entity categories change frequently [20,21]. Metric learning has become one of the dominant paradigms in few-shot learning [17–19]. Prototypical networks learn a representation space in which each class is represented by a prototype, and query instances are classified according to their distances to these prototypes [22]. This idea provides a simple and scalable inductive bias for low-resource classification and has been widely adapted to few-shot NER.

In few-shot NER, metric-based methods usually employ pretrained language models, such as BERT, to obtain contextual token or span representations [13,14]. The class prototype is then constructed from support examples, and query tokens or spans are classified through nearest-prototype matching. StructShot demonstrated that a nearest-neighbor

classifier combined with structured decoding can serve as a strong baseline for few-shot NER [23]. Label-agnostic nearest-neighbor transfer and explicit label-semantic modeling have further broadened this metric-learning direction [32,33]. CONTaiNER introduced contrastive learning over Gaussian token embeddings to improve cross-domain generalization [24], while prompt-guided contrastive learning and coarse-to-fine label modeling have explored richer label semantics for few-shot transfer [34,35]. These methods improve the discriminability of the metric space, but they still mainly rely on support samples to construct prototypes. When the number of support examples is extremely small, the resulting prototypes may be biased, incomplete, or unstable. Moreover, fine-grained semantic relationships among entity types are not fully exploited.

2.2. Decomposed and Span-Based Few-Shot NER

Most early few-shot NER methods follow a token-wise sequence labeling paradigm. However, token-wise prediction is easily affected by the dominant non-entity class and may fail to model explicit mention boundaries. This problem is particularly serious in low-resource settings, where a small number of labeled examples is insufficient for learning reliable boundary patterns [20,21]. To alleviate this issue, decomposed and span-based methods have been proposed to separate entity localization from entity type classification [25–27].

DecomposedMetaNER sequentially models span detection and entity typing through a meta-learning framework, showing that decomposing the task can reduce the difficulty of few-shot NER [25]. SpanProto further transforms sequence labeling into a global boundary matrix and performs span-level prototypical classification [26]. Subsequent studies have extended span decomposition through enhanced multi-stage architectures and explicit boundary-discrimination or correlation-purification mechanisms [36–38]. More recently, TadNER identified two persistent problems in decomposed few-shot NER systems: over-detection of false spans and inaccurate or unstable prototypes. It addresses these issues through type-aware span filtering and type-aware contrastive learning [27]. These studies indicate that explicit span modeling is beneficial for few-shot NER, especially when entity boundaries are ambiguous. Nevertheless, most existing span-based methods still construct prototypes mainly from support spans or type names, without sufficiently considering structured semantic knowledge among entity types.

2.3. Knowledge-Enhanced and LLM-Assisted Few-Shot NER

Knowledge-enhanced NER methods introduce external semantic information to enrich entity and type representations. In few-shot settings, such knowledge is particularly useful because support examples are often too limited to provide complete class semantics. KCL is a representative knowledge-enhanced few-shot NER method that combines knowledge graphs with contrastive learning. Its main objective is to improve the discriminability of the representation space by pulling semantically related entity representations closer and pushing unrelated representations apart. Therefore, KCL can improve few-shot NER by using graph knowledge as an auxiliary representation-learning signal, but it still follows the encoder-based prototype learning paradigm and focuses primarily on representation separation rather than span boundary correction or sample-size-aware prototype reliability [28]. In biomedical NER, knowledge-guided instance generation has also shown that graph relations can help enrich scarce training samples and improve entity recognition performance [39]. These studies suggest that structured label and entity knowledge can serve as an effective semantic prior for low-resource NER [1–3].

KGen more directly investigates what knowledge graphs can contribute to few-shot NER [29]. It introduces KG-derived class features and injects knowledge from both the

input-sentence side and the label-semantic side, making it more systematic than methods that use KG information only as auxiliary representation regularization. This design is closely related to the label semantic enrichment problem addressed in this paper. However, KGen mainly emphasizes knowledge injection for class-feature construction and semantic matching, whereas KG-APC further couples KG-derived type priors with boundary-aware span detection and adaptive prototype correction. In other words, KG-APC is designed to handle two industrial few-shot NER bottlenecks simultaneously: incomplete or noisy entity boundaries and unstable prototypes produced from very limited support spans. Recent knowledge-enhanced NER studies, such as Know-Adapter [40] and SCANNER [41], also show that external knowledge can be injected through adapters, graph-aware representation learning, and multi-modal semantic alignment, but they do not explicitly model the sample-size-dependent balance between support evidence and KG priors in fine-grained industrial NER.

Recent large language model-assisted methods provide another direction for improving few-shot NER. For example, Double-Checker and BANER use large language models to verify candidate spans or enhance boundary-aware few-shot extraction [42,43]. PromptNER also shows how entity definitions and demonstrations can guide LLM-based few-shot extraction [44], building on the broad few-shot capability of large language models [45]. These approaches show strong potential because large language models contain rich general and domain knowledge. However, they may require higher computational cost, careful prompt design, and additional inference procedures. Therefore, a compact and trainable knowledge-guided metric model remains valuable, especially for industrial maintenance information systems where deployment efficiency and annotation cost are important considerations.

For industrial fault diagnosis, Xu et al. developed VOC-LLM to couple multidimensional vibration features and variable-speed information with a lightly adapted large language model, thereby supporting interpretable fault-type recognition and quantitative severity assessment under variable operating conditions [6]. Their laboratory and industrial-platform evaluations demonstrate the potential of domain-adapted large models for predictive maintenance. VOC-LLM and KG-APC nevertheless address different modalities and task levels: VOC-LLM performs diagnosis from operating signals, whereas KG-APC extracts fine-grained entities from maintenance text under few-shot supervision. The two directions are complementary because structured equipment, fault, cause, and maintenance-action entities can support the textual knowledge layer used alongside signal-based diagnosis and downstream maintenance decision support.

Different from existing methods, the proposed KG-APC follows the two-stage philosophy of decomposed few-shot NER but focuses on knowledge-guided prototype correction. Instead of relying only on type names or limited support spans, KG-APC obtains graph-aware type priors from the structural neighborhoods of entity types and uses them to adaptively correct episode-level prototypes. Compared with KCL, KG-APC does not use the KG only to regularize the embedding space through contrastive learning; compared with KGen, it does not only inject KG-derived class features from the input and label sides. Instead, it explicitly uses the KG prior to correct support-set prototypes through a K-aware gate after boundary-aware span detection has produced cleaner candidate mentions. In this way, the model jointly addresses boundary detection errors and prototype instability under scarce supervision, while maintaining a lightweight metric-learning framework for few-shot industrial NER.

3. Knowledge Graph-Guided Adaptive Prototype Correction Framework

3.1. Problem Definition

A few-shot NER task is formulated as a sequence of episodic recognition problems. Each episode contains an entity type set $\mathcal{C} = \{c_1, c_2, \dots, c_N\}$, a support set $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^{N \times K}$, and a query set $\mathcal{Q} = \{x_j^q\}_{j=1}^M$. Here, $\mathcal{C}$ denotes the episode-specific label space, c_i is the i -th entity type, and N is the number of entity types in the episode. An episode with N entity types and K labeled support examples for each type is called an N -way K -shot episode. In the support set, x_i^s denotes the i -th support sentence or text instance, y_i^s denotes its gold entity annotation, and the superscript s indicates that the example belongs to the support set. Similarly, x_j^q denotes the j -th query instance, the superscript q indicates that it belongs to the query set, and M is the number of query instances. For a query sentence $X = (x_1, \dots, x_n)$, the model outputs a set of entity predictions $\hat{\mathcal{Y}} = \{(b_m, e_m, c_m)\}$, where b_m and e_m are the start and end boundaries of the m -th predicted entity span and $c_m \in \mathcal{C}$ is its predicted entity type. Spans that do not correspond to entities are rejected by the boundary detector or by a confidence threshold.

In addition to the support and query examples, KG-APC uses Wikidata as the external knowledge graph $G = (\mathcal{V}, \mathcal{R}, \mathcal{A})$ to provide structured semantic priors for entity types [3]. The node set $\mathcal{V}$ contains entity type nodes and may also include entity nodes, attribute nodes, or other concept nodes retrieved from Wikidata. The relation set $\mathcal{R}$ describes semantic links among these nodes, such as instance-of, subclass-of, related-to, and attribute-oriented relations. The auxiliary information set $\mathcal{A}$ contains textual aliases, descriptions, or type definitions associated with Wikidata nodes. Under this setting, the learning objective is to learn the conditional distribution of entity predictions given the query sentence, the support set, and the external graph, namely, $P(\mathcal{Y} | X, \mathcal{S}, G)$, and to select the prediction set that best matches the entity boundaries and types in the query text.

The KG-APC framework consists of three main components: a boundary-aware span detector, a dynamic prototype constructor with knowledge graph priors, and a metric-based classifier. The boundary-aware span detector first generates candidate mentions without assigning fine-grained entity types. The dynamic prototype constructor then builds an initial textual prototype for each entity class and corrects it using structured semantic priors from a knowledge graph. Finally, the metric-based classifier compares each query span with the corrected class prototypes and assigns the most likely entity type. The overall architecture of KG-APC is illustrated in Figure 1.

3.2. Boundary-Aware Span Detection

The span detector enumerates candidate intervals $[i, j]$ for a sentence $X = [x_1, \dots, x_n]$, where X is the input token sequence, x_t is the t -th token, and n is the sentence length. The indices i and j denote the start and end positions of a candidate span, respectively, with $1 \leq i \leq j \leq n$. For instance, in the MaintIE maintenance short-text example “change out engine”, the phrase “change out” forms a complete maintenance-activity span and “engine” forms a physical-object span, whereas the partial token “change” alone does not preserve the complete activity mention [4]. This boundary distinction is especially important for few-shot NER in industrial maintenance data, because maintenance records are often short, elliptical, and written with domain-specific action–object expressions. If the model directly assigns entity labels to individual tokens, it may split multi-token actions or merge adjacent equipment and fault descriptions into incorrect entities. By detecting complete candidate spans before type classification, the model can provide cleaner entity candidates and reduce boundary-induced classification errors under scarce annotations.

First, we use a pretrained Transformer encoder, such as BERT [13] or RoBERTa [14], to obtain contextual token representations $[h_1, \dots, h_n] = \text{Enc}_{\text{text}}(x_1, \dots, x_n)$. Here, Enc_{text}

denotes the text encoder and h_t is the contextual embedding of token x_t . Each span representation s_{ij} concatenates the start token, end token, pooled internal context, and trainable boundary and length embeddings, following the motivation of explicit span modeling in few-shot NER [25–27].

$$s_{ij} = \text{MLP}_s(h_i \parallel h_j \parallel \text{Pool}(h_{i:j}) \parallel w_{j-i+1} \parallel b_i \parallel b_j). \quad (1)$$

In this equation, s_{ij} is the final representation of the candidate span from position i to position j . The function MLP_s is a multilayer perceptron used to project the concatenated span features into the span representation space. The vectors h_i and h_j represent the contextual embeddings of the start and end tokens, while $\text{Pool}(h_{i:j})$ denotes a pooling operation over all token embeddings inside the span. The symbol $\parallel$ denotes vector concatenation. The term w_{j-i+1} is a trainable length embedding indexed by the span length $j - i + 1$, and b_i and b_j are trainable boundary-position embeddings associated with the start and end positions.

KG-APC Framework for Few-Shot Industrial Named Entity Recognition

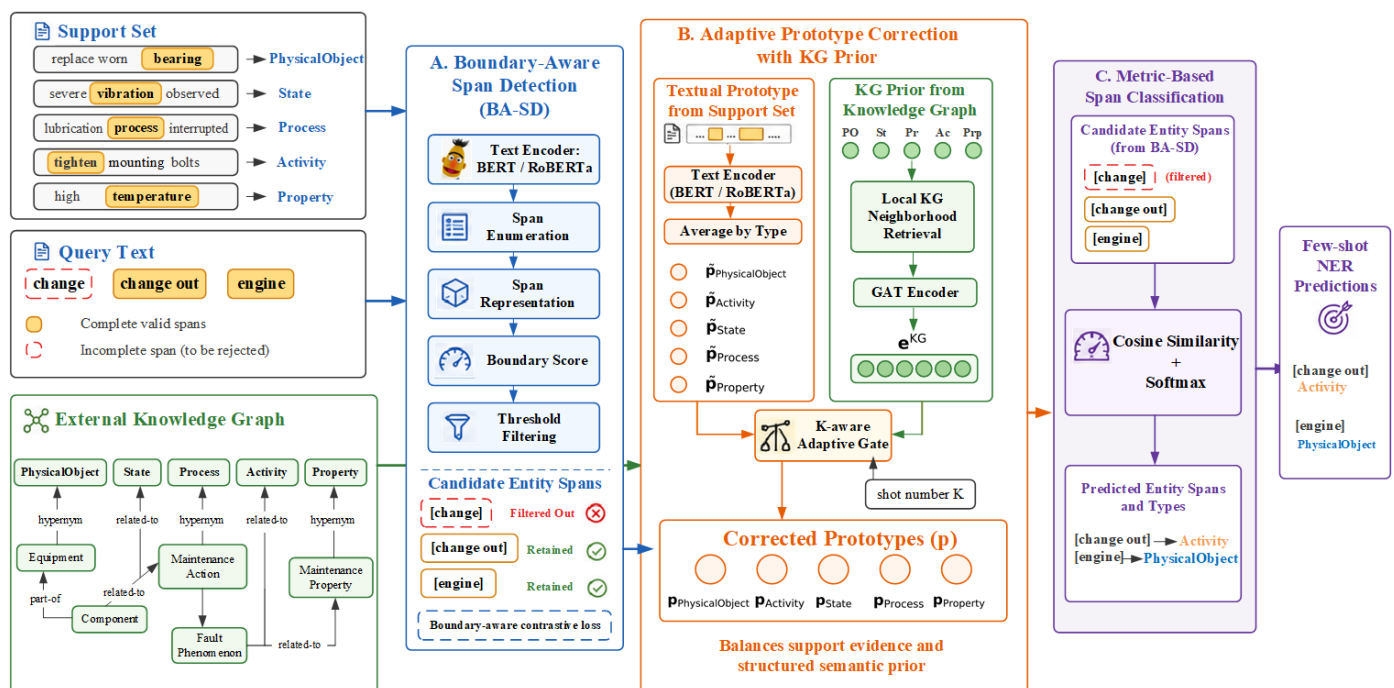


Figure 1. Architecture of KG-APC with boundary-aware span detection and adaptive prototype correction.

After obtaining the span representation, the detector needs to decide whether the candidate interval $[i, j]$ corresponds to a complete entity mention rather than a partial phrase or a non-entity fragment. We therefore apply a binary boundary scoring layer to s_{ij} . This layer compresses the span-level representation into a scalar confidence score, where a larger value indicates that the start and end positions are more likely to form a valid entity boundary. The boundary score is computed as follows:

$$B_{ij} = \text{sigmoid}(w_b s_{ij} + b_b). \quad (2)$$

The scalar B_{ij} denotes the boundary confidence score of span $[i, j]$. The vector w_b and scalar b_b are the weight and bias parameters of the boundary scoring layer, respectively.

The sigmoid function maps the raw boundary score into the interval $[0, 1]$, so that B_{ij} can be interpreted as the probability that span $[i, j]$ corresponds to a valid entity mention.

After all candidate intervals have been assigned boundary scores, the detector filters them using a fixed threshold. This step converts the dense set of enumerated intervals into a smaller set of candidate entity spans that will be passed to the subsequent type classification module. The candidate span set is defined as follows:

$$\mathcal{S}_{\text{cand}} = \{[i, j] \mid B_{ij} \geq \tau_b\}. \quad (3)$$

The set $\mathcal{S}_{\text{cand}}$ contains all candidate spans whose boundary confidence scores are no smaller than the threshold τ_b . Thus, τ_b controls the trade-off between retaining more potential entity mentions and filtering out false spans at the boundary detection stage.

Since gold entity annotations are available in the training set, we construct positive spans s_p and negative spans s_n for boundary representation learning. Specifically, positive spans are defined as spans that exactly match the gold entity boundaries, while negative spans are generated by random sampling or by selecting spans with inappropriate lengths, such as overly long or overly short spans. To make the span encoder sensitive to boundary correctness, we organize the training spans into triplets and optimize a margin-based contrastive objective. This objective pulls an anchor span closer to a correct-boundary span and pushes it away from an incorrect-boundary span [24,46]. For each training triplet, a denotes an anchor span representation, s_p denotes a positive span representation with a correct boundary, and s_n denotes a negative span representation with an incorrect boundary.

$$\mathcal{L}_{\text{BA}} = \sum_{(a, s_p, s_n)} \max(0, d(a, s_p) - d(a, s_n) + m_b). \quad (4)$$

Here, $\mathcal{L}_{\text{BA}}$ denotes the boundary-aware contrastive loss. The distance function $d(\cdot, \cdot)$ measures the distance between two span representations in the embedding space, and m_b is the margin that enforces a minimum separation between positive and negative boundary pairs. The hinge term $\max(\cdot)$ penalizes cases in which the anchor span is not sufficiently closer to the positive span than to the negative span. By minimizing this loss, the detector learns boundary representations that distinguish correct entity spans from noisy or incomplete candidate spans.

3.3. Adaptive Prototype Correction with KG Prior

Prototype learning is a natural choice for few-shot NER because it converts each entity type into a representative vector and classifies query spans by their distance to these type representations. However, in industrial maintenance scenarios, a prototype constructed only from a few support mentions is often unstable. Maintenance texts contain many abbreviated actions, equipment names, fault descriptions, and component aliases, and different episodes may sample very different mentions for the same entity type. As a result, a purely textual prototype can be biased toward the accidental wording of the current support set rather than the stable semantics of the entity type.

The core idea of Adaptive Prototype Correction with KG Prior is to treat Wikidata as a semantic anchor for prototype construction. Instead of replacing the support-set evidence, the KG prior provides complementary information about the entity type, including aliases, textual descriptions, parent concepts, related concepts, and representative instances. This is particularly useful for low-shot industrial NER, where the support set may contain only one or a few mentions of a type, but Wikidata can still provide broader semantic context for concepts such as equipment, components, maintenance actions, fault phenomena, or diagnostic attributes. In this way, knowledge graph enhancement changes the role

of prototypes from simple averages of scarce examples to corrected representations that combine episode-specific textual evidence with more stable type-level semantics.

In this paper, the external KG is implemented with Wikidata [3]. For each entity type in the episode, we link the type name and its available aliases or descriptions to Wikidata items, and we then retrieve a local semantic neighborhood from the linked item. This unified KG source is used for both general few-shot NER benchmarks and industrial maintenance datasets. For MaintIE and CFDK, entity schema labels such as physical objects, activities, equipment, faults, causes, and troubleshooting solutions are aligned to Wikidata through type names, aliases, and short descriptions, so that the KG encoder can retrieve structured semantic context for each target entity type [4,5]. Therefore, the knowledge prior used by KG-APC is not an unrelated external signal but a structured representation derived from Wikidata concepts that are semantically aligned with the label space to be recognized.

To make the KG component reproducible, we use a deterministic type-to-Wikidata alignment protocol. For each entity type, the label name, aliases, and short definition are first normalized by lowercasing, lemmatization, and removal of punctuation. Candidate Wikidata items are then retrieved by exact label matching, alias matching, and description matching. When multiple items are retrieved, priority is given to items whose descriptions are semantically closest to the dataset definition and whose local neighborhood contains maintenance-, equipment-, state-, process-, or property-related concepts. If no reliable item is found, the type is linked to the nearest superclass concept and marked as a weakly aligned type. The retrieved neighborhood is restricted to one-hop and two-hop relations used in Wikidata, including subclass-of, instance-of, part-of, has-part, attribute-oriented relations, and semantically related claims. To avoid introducing excessive noise, each type retains the top-ranked neighbors after duplicate removal and relation filtering. Node features are initialized by encoding the concatenation of the item label, aliases, and description with the same text encoder used for span representation, and relation labels are used as edge-type indicators in the graph encoder. Representative type-to-Wikidata alignments and their retained two-hop neighborhoods are summarized in Table 1.

For the MaintIE fine-grained setting, we use the 32-class intermediate granularity defined by the MaintIE schema, consisting of the 5 top-level classes and their 27 second-level subclasses. To keep the core method description concise, Table 2 presents a representative excerpt spanning the 5 top-level branches, while the complete mapping of all 32 types is provided in Supplementary Table S1. The neighborhood size remains consistent with the experimental configuration: each type retrieves a relation-filtered two-hop Wikidata neighborhood.

The correction must also be adaptive. When the episode is extremely low-shot, the KG prior should contribute more because the support-set prototype is noisy; when more support examples are available, the textual prototype becomes more reliable and should dominate the final representation. KG-APC therefore introduces a sample-size-aware gate to learn how much information should be taken from the support examples and how much should be taken from the KG prior for each entity type. This design preserves the flexibility of metric-based few-shot learning while reducing prototype bias in sparse industrial annotation settings.

Table 1. Examples of type-to-Wikidata alignment and representative two-hop neighborhoods used for KG prior construction.

Dataset	Entity Type	Wikidata Anchor Concept	Representative Retained Two-Hop Neighborhood
MaintIE coarse	PhysicalObject	physical object/equipment	equipment, machine, component, part-of relation
MaintIE coarse	State	state/condition	condition, operating state, diagnostic state, fault state
MaintIE coarse	Process	process/event	operation, temporal event, functional process
MaintIE coarse	Activity	activity/maintenance	maintenance, task, action, work order
MaintIE coarse	Property	property/physical property	attribute, temperature, pressure, vibration
CFDK	Equipment	equipment	compressor, subsystem, component, part-of relation
CFDK	Fault	fault/failure	malfunction, fault phenomenon, abnormal state, failure process
CFDK	Service Request	service request/maintenance request	maintenance, repair request, work order, service action
CFDK	Cause	cause/reason	fault cause, diagnostic explanation, failure mechanism, reason
CFDK	Troubleshooting Solution	troubleshooting/repair	corrective action, repair procedure, maintenance action, solution

Table 2. Representative excerpt of the MaintIE fine-grained entity mapping and retained two-hop Wikidata neighborhoods. The complete 32-type mapping is provided in Supplementary Table S1.

Representative MaintIE Type	Wikidata Anchor Concept	Representative Retained Two-Hop Neighborhood
Activity/MaintenanceActivity	maintenance/repair activity	repair, replacement, inspection, calibration
Activity/SupportingActivity	supporting activity/operation	operation, measurement, isolation, assembly
PhysicalObject/SensingObject	sensor/measuring instrument	pressure sensor, temperature sensor, flow sensor, diagnostic device
PhysicalObject/DrivingObject	drive/actuator	motor, engine, actuator, drive unit
State/UndesirableState	fault state/abnormal condition	failed, degraded, abnormal, seized
Process/UndesirableProcess	undesirable process/failure process	leakage, overheating, wear, interruption
Property/DesirableProperty	desirable property/acceptable attribute	nominal value, acceptable pressure, normal temperature
Property/UndesirableProperty	undesirable property/abnormal attribute	excessive vibration, high temperature, low pressure

Following this motivation, we first construct a textual prototype from the support set and then correct it with a graph-derived type prior. For each entity type c , we extract the representations of all gold entity spans of this type from the support set and compute their mean representation as the initial textual prototype $\tilde{p}_c$. Let $\mathcal{S}_c$ denote the subset of support examples or support spans labeled as entity type c , and let $|\mathcal{S}_c|$ denote the number of such labeled spans. For each pair $(x, y) \in \mathcal{S}_c$, x denotes a support text instance and y denotes the corresponding gold entity span or entity annotation. The function $h_{\text{span}}(x, y)$ extracts the

span representation associated with annotation y in input x . This prototype summarizes the gold entity representations of type c in the support set:

$$\tilde{p}_c = \frac{1}{|\mathcal{S}_c|} \sum_{(x,y) \in \mathcal{S}_c} h_{\text{span}}(x, y). \quad (5)$$

The vector $\tilde{p}_c$ is therefore the support-set-based textual prototype of class c . It is computed by averaging the span representations of all available support mentions of that class. Because K is small in few-shot learning, $\tilde{p}_c$ may be sensitive to the specific examples sampled in the episode.

Since only a few support examples are available in the K -shot setting, the initial textual prototype $\tilde{p}_c$ may suffer from sample selection bias, contextual shift, and entity ambiguity. To mitigate these problems, we introduce a KG encoder to derive the structured prior representation e_c^{KG} for the same entity type. For each entity type node c linked to Wikidata, we retrieve its one-hop or two-hop neighborhood $N(c)$, which consists of superclass concepts, related items, attributes, representative instances, and relation nodes. The neighbor nodes of entity type c are denoted as $v \in N(c)$. In the graph encoder, $h_c^{(l)}$ denotes the representation of the central entity type node c at layer l , and $h_v^{(l)}$ denotes the representation of a neighboring node v at the same layer. We then adopt a graph attention network [47] to adaptively aggregate the neighboring information and obtain the structured knowledge prior e_c^{KG} for entity type c , consistent with prior evidence that knowledge graphs can improve label semantic modeling under low-resource NER settings [28,39]. The first step of this graph encoding process is to compute how relevant each neighboring node is to the central entity type node:

$$e_{cv}^{(l)} = \text{LeakyReLU}\left(a^{(l)\top} \left[W^{(l)} h_c^{(l)} \| W^{(l)} h_v^{(l)} \right]\right). \quad (6)$$

The value $e_{cv}^{(l)}$ is the unnormalized attention score between the central entity type node c and its neighboring node v at GAT layer l . The matrix $W^{(l)}$ is the layer-specific linear transformation applied to node features, and $a^{(l)}$ is the trainable attention vector at layer l . The superscript $\top$ denotes vector transpose. The LeakyReLU function introduces nonlinearity before attention normalization, allowing the model to assign different importance values to different semantic neighbors.

The raw scores $e_{cv}^{(l)}$ are not directly comparable across neighborhoods until they are normalized. Therefore, for each central node c , we apply a softmax operation over its neighboring nodes so that the model obtains a distribution of attention weights for knowledge aggregation:

$$\alpha_{cv}^{(l)} = \text{softmax}_v \left(e_{cv}^{(l)} \right). \quad (7)$$

The coefficient $\alpha_{cv}^{(l)}$ is the normalized attention weight from node c to node v at layer l . The operator softmax_v normalizes the scores over all neighboring nodes v of the same central node c , so that more informative neighbors receive larger aggregation weights.

After the attention weights have been obtained, the KG encoder aggregates the transformed representations of neighboring nodes. This produces a graph-derived type prior that summarizes the structural context of entity type c :

$$e_c^{\text{KG}} = \sigma \left(\sum_{v \in N(c)} \alpha_{cv} W h_v \right). \quad (8)$$

In this aggregation step, $\mathcal{N}(c)$ denotes the neighborhood of entity type node c in the knowledge graph and is used consistently with $N(c)$. The term α_{cv} denotes the final attention weight assigned to neighbor v , W is the graph-level projection matrix, and h_v is the final representation of neighbor node v . The activation function $\sigma(\cdot)$ maps the aggregated neighbor information into the structured prior representation e_c^{KG} . This vector encodes semantic information about type c that is not necessarily observable from the limited support examples.

We further design a sample-size-aware gating mechanism g_c . Intuitively, when K is small, the model should rely more on the KG prior; when K becomes larger, the support-set prototype becomes more reliable. We use g_c exclusively for the class-specific fusion gate, while α_{cv} remains the graph attention coefficient from node c to neighbor v . Before computing the gate, we encode the shot number into a bounded scalar so that the model can use the amount of support evidence as an explicit input:

$$\bar{k} = \tanh\left(\frac{K}{K_{\max}}\right). \quad (9)$$

Here, $\bar{k}$ is a normalized scalar representation of the shot number. The value K is the number of labeled support examples per class, and $K_{\max}$ is the maximum shot number considered in the training or evaluation setting. We set $K_{\max} = 10$, matching the largest shot value used in the reported industrial experiments. The hyperbolic tangent function $\tanh(\cdot)$ maps the ratio $K/K_{\max}$ into a bounded range, allowing the gate to perceive the relative amount of support evidence available in the current episode.

Given the textual prototype, the KG prior, and the normalized shot indicator, the model computes a class-specific fusion weight. This gate determines whether the corrected prototype should be closer to the support-set evidence or to the structured graph prior:

$$g_c = \sigma\left(W_g \left[\tilde{p}_c \parallel e_c^{\text{KG}} \parallel \bar{k}\right] + b_g\right). \quad (10)$$

In the gate computation, W_g and b_g are trainable parameters of the gating layer. The input to the gate is the concatenation of the initial textual prototype $\tilde{p}_c$, the KG prior e_c^{KG} , and the normalized shot indicator $\bar{k}$. The sigmoid function $\sigma(\cdot)$ produces the gate value g_c , which controls how much the final class prototype should rely on support-set evidence rather than structured prior knowledge.

Finally, KG-APC uses the gate value to interpolate between the two sources of class information. The support-set prototype provides episode-specific textual evidence, whereas the KG prior provides more stable semantic structure. The corrected prototype is obtained by the following weighted fusion:

$$p_c = g_c \tilde{p}_c + (1 - g_c) e_c^{\text{KG}}. \quad (11)$$

The vector p_c is the corrected prototype of entity type c . When g_c is close to 1, the corrected prototype is dominated by the textual prototype $\tilde{p}_c$; when g_c is close to 0, it relies more on the KG prior e_c^{KG} . This adaptive interpolation allows KG-APC to reduce prototype bias in extremely low-shot cases while preserving data-driven evidence when more support examples are available.

3.4. Metric-Based Classification

After obtaining the refined prototype p_c for each entity type, we classify query spans by comparing them with class prototypes in the embedding space. Each candidate span s_{ij} is treated as a query instance, and each refined prototype p_c represents entity type c . Cosine similarities between s_{ij} and all prototypes are normalized by a softmax function to obtain

the type distribution, following the general metric-classification principle of prototypical few-shot learning [22]:

$$P(c \mid s_{ij}) = \text{softmax}_c(\cos(s_{ij}, p_c)). \quad (12)$$

The term $P(c \mid s_{ij})$ denotes the predicted probability that candidate span $[i, j]$, represented by s_{ij} , belongs to entity type c . The function $\cos(s_{ij}, p_c)$ computes the cosine similarity between the query span representation and the corrected prototype of class c . The operator softmax_c normalizes similarities over all candidate entity types in $\mathcal{C}$, producing a probability distribution over the episode-specific label space.

The overall loss consists of the type classification loss and the boundary-aware contrastive loss. Let $\mathcal{D}_{\text{cls}}$ denote the set of candidate spans used for supervised type classification during training. For each candidate span $(i, j) \in \mathcal{D}_{\text{cls}}$, y_{ij}^c is a binary indicator that equals 1 if span $[i, j]$ belongs to class c and 0 otherwise. To train the metric classifier, we compare the predicted type distribution with the gold span label and minimize the span-level cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{|\mathcal{D}_{\text{cls}}|} \sum_{(i,j) \in \mathcal{D}_{\text{cls}}} \sum_{c \in \mathcal{C}} y_{ij}^c \log P(c \mid s_{ij}). \quad (13)$$

The loss $\mathcal{L}_{\text{cls}}$ is the cross-entropy loss for span-level entity type classification. The denominator $|\mathcal{D}_{\text{cls}}|$ normalizes the loss by the number of supervised candidate spans, and the inner summation over $c \in \mathcal{C}$ accumulates the classification error across all entity types. The logarithm term $\log P(c \mid s_{ij})$ penalizes the model when it assigns low probability to the correct class.

The final training step combines type discrimination and boundary representation learning. This joint objective allows the model to improve span localization and entity type assignment at the same time:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_{\text{BA}} \mathcal{L}_{\text{BA}}. \quad (14)$$

The final training objective $\mathcal{L}$ combines the classification loss $\mathcal{L}_{\text{cls}}$ and the boundary-aware contrastive loss $\mathcal{L}_{\text{BA}}$. The hyperparameter λ_{BA} controls the relative contribution of boundary representation learning. A larger λ_{BA} places greater emphasis on boundary discrimination, whereas a smaller value makes the model focus more on entity type classification.

3.5. Computational Efficiency Analysis

To characterize the computational requirements of KG-APC before presenting the empirical efficiency comparison, we analyze its time and space complexity. The corresponding measurements of inference latency, FPS, FLOPs, parameter count, training time, memory usage, and Wikidata retrieval cost are reported later in Section 4.5. Let n denote the input sentence length, L_{max} denote the maximum candidate span length, C denote the number of entity types in an episode, and d denote the hidden dimension. In the boundary-aware span detector, candidate spans are enumerated under a maximum-length constraint, and therefore the number of candidate spans is bounded by

$$|\mathcal{S}_{\text{cand}}| \leq nL_{\text{max}}. \quad (15)$$

This reduces the span enumeration cost from the unconstrained $O(n^2)$ scale to $O(nL_{\text{max}})$. The span scoring stage has complexity $O(nL_{\text{max}}d)$, and the metric classification stage compares each retained span with C corrected class prototypes, resulting in a complexity of

$$O(|\mathcal{S}_{\text{cand}}|Cd) = O(nL_{\text{max}}Cd). \quad (16)$$

The KG encoder aggregates local type-neighborhood information from Wikidata. If the graph aggregation is performed online, its asymptotic complexity is $O(L_g |E_G^{ep}| d)$, where L_g is the number of GAT layers and $|E_G^{ep}|$ denotes the number of edges in the episode-specific local KG. In our implementation, Wikidata neighborhoods are retrieved and cached before episodic training and inference. Therefore, the online inference stage only needs to load cached type priors and perform adaptive prototype correction, reducing the practical KG-related inference cost to $O(Cd)$. Considering the RoBERTa encoder, the overall inference complexity of KG-APC can be written as

$$O(L_e n^2 d + L_e n d^2 + n L_{\max} d + L_g |E_G^{ep}| d + n L_{\max} C d), \quad (17)$$

where L_e is the number of Transformer layers. With cached KG priors, the practical online complexity becomes

$$O(L_e n^2 d + L_e n d^2 + n L_{\max} C d). \quad (18)$$

Thus, KG-APC remains in the same complexity class as two-stage span-based prototype methods, while introducing only a small and controllable KG-prior lookup and prototype-correction overhead.

The space complexity of KG-APC is

$$O(P_\theta + n d + |\mathcal{S}_{\text{cand}}| d + C d + |V_G^{ep}| d), \quad (19)$$

where P_θ denotes the model parameters and $|V_G^{ep}|$ denotes the number of local KG nodes used in the current episode. After KG-prior caching, the online space complexity is reduced to

$$O(P_\theta + n d + n L_{\max} d + C d). \quad (20)$$

This analysis predicts a substantially smaller online time and space footprint than LLM-assisted methods such as BANER [43]. Although BANER uses LoRA for parameter-efficient tuning, its inference pipeline still loads a LLaMA-2-7B-scale backbone, whereas KG-APC uses RoBERTa-base and cached type priors. The empirical comparison in Section 4.5 tests whether this structural difference translates into lower latency, fewer FLOPs, and lower peak memory under the same hardware environment.

4. Experiments and Results

The experiments were designed to answer four questions. First, we examined whether KG-APC improves few-shot NER on industrial maintenance data, where entity mentions are short, domain-specific, and often difficult to delimit. Second, we compared KG-APC with representative few-shot NER baselines on general benchmarks to test whether the method remains competitive outside the industrial domain. Third, we measured whether the compact encoder and offline KG caching provide practical time- and space-efficiency advantages over the LLM-assisted BANER baseline. Fourth, we conducted ablation and sensitivity analyses to isolate the contributions of the knowledge graph prior, the adaptive gate, and boundary-aware span detection. The evaluation therefore covers MaintIE, CFDK, Few-NERD INTER, Few-NERD INTRA [30], and a fixed- K few-shot setting built from CoNLL-2003 [4,5,31].

4.1. Datasets and Task Settings

MaintIE is an industrial maintenance information extraction benchmark built from maintenance short texts [4]. It contains 1076 multi-annotator fine-grained annotated texts and provides a hierarchical entity schema for maintenance records. We construct two few-shot NER settings from MaintIE. The first is a coarse-grained NER setting, where

each episode is a 5-way task over the five top-level MaintIE classes: PhysicalObject, State, Process, Activity, and Property. This setting evaluates whether a model can recognize broad maintenance entity categories with very limited support examples. The second is a fine-grained NER setting, where each episode samples entity types from 32 intermediate fine-grained MaintIE types. This setting is more challenging because fine-grained types are semantically closer to each other and because short maintenance texts often contain compact action-object and fault-state expressions.

CFDK is a compressor fault diagnosis knowledge extraction dataset constructed from maintenance log sheets [5]. It contains 6196 valid text pairs and was annotated by three domain experts. The annotation schema includes five entity or sequence categories: Equipment, Fault, Service Request, Cause, and Troubleshooting Solution. Because CFDK has exactly five target categories in our few-shot formulation, we evaluate it under 5-way 1-shot, 5-way 5-shot, and 5-way 10-shot settings. However, the 5-way label space does not make CFDK a simple dataset. The difficulty differs substantially across categories: Equipment mentions are usually shorter and more regular, whereas Cause and Troubleshooting Solution often appear as long descriptive sequences with complex boundaries. The original fully supervised CFDK results also show that these harder categories obtain clearly lower recognition performance than easier equipment-related categories [5]. The dataset complements MaintIE by focusing on compressor fault diagnosis rather than general maintenance short texts, which allows us to test whether KG-APC transfers across different industrial maintenance scenarios.

MaintIE and CFDK, the two industrial datasets used in this study, are publicly available research datasets rather than private collections. Their original dataset papers provide the corresponding public repository addresses and access information [4,5]. We use only these publicly released data and do not use IMER or any private industrial subset. To make the evaluated industrial examples and model decisions more directly inspectable, Section 4.8 further presents a MaintIE case study with the query text, enumerated candidate spans, boundary scores, prototype decisions before and after KG correction, and gold fine-grained labels.

Few-NERD provides a two-level fine-grained entity taxonomy and is widely used for few-shot NER evaluation [30]. The INTER split evaluates transfer to unseen entity types across coarse domains, whereas the INTRA split evaluates fine-grained discrimination within the same coarse-grained domain. We report 5-way 1-shot, 5-way 5-shot, 10-way 1-shot, and 10-way 5-shot results for Few-NERD. CoNLL-2003 contains four coarse-grained English news-domain entity types, PER, LOC, ORG, and MISC, and it is used as a supplementary fixed- K benchmark [31]. Since CoNLL-2003 has four entity categories, it is evaluated as a 4-way setting with $K = 1$, $K = 5$, and $K = 10$.

For all episodic experiments, an N -way K -shot episode contains N sampled entity types and K labeled support examples for each type. We use 5-way 1-shot, 5-way 5-shot, and 5-way 10-shot settings for the industrial datasets where the label space supports them. For datasets or label spaces with more than five available categories, including MaintIE fine-grained and Few-NERD, we further report 10-way settings to evaluate model robustness under a larger episode label space. In MaintIE fine-grained, this includes 10-way 1-shot, 10-way 5-shot, and 10-way 10-shot episodes. Micro-F1 is used as the main evaluation metric because it jointly reflects entity boundary detection and entity type classification performance at the corpus level.

For each locally evaluated dataset, support and query examples are sampled episode by episode from non-overlapping splits. The same sampled episodes are reused across locally evaluated models in a given setting to reduce sampling-induced variation. For Few-NERD baselines quoted from the BANER comparison protocol, we follow the published

episode setting instead of claiming local resampling. Five random seeds are used for episode construction and model initialization in local experiments. The validation split is used for early stopping and hyperparameter selection, while the test split is used only for final reporting. For the industrial datasets, MaintIE coarse-grained and CFDK are evaluated under 5-way episodes because their target label spaces contain five top-level categories, whereas MaintIE fine-grained additionally supports 10-way episodes because it contains 32 intermediate fine-grained types.

4.2. Baselines

The compared methods cover the main NER model categories discussed in the related work. ProtoBERT is a pretrained language model-based metric learning baseline. It constructs class representations from support examples and classifies query tokens or spans by similarity to those prototypes [20]. This model represents the prototypical-network family of few-shot NER methods.

CONTaiNER belongs to the contrastive few-shot NER category. It improves the embedding space by contrastive learning over contextual representations, aiming to make same-type entities more compact and different-type entities more separable under domain transfer [24]. DecomposedMetaNER represents decomposed few-shot NER methods. It separates entity span detection from entity type classification, reducing the difficulty of directly predicting fine-grained labels in low-resource episodes [25]. SpanProto is a span-based metric learning baseline that explicitly represents candidate entity spans and compares them with class prototypes [26]. It is especially relevant to KG-APC because both methods treat entity spans as the basic classification units rather than relying only on token-level labeling.

TadNER is a task-decomposed few-shot NER model that further explores the separation between boundary identification and type classification [27]. KCL and KGen are included as knowledge-enhanced few-shot NER baselines. KCL represents the KG-plus-contrastive-learning family: it uses knowledge graphs and contrastive objectives to make entity representations more discriminative. KGen is a stronger and more systematic KG-based baseline because it uses KG-derived class features and injects knowledge from both the input sentence side and the label semantic side. These two baselines are important for testing whether KG-APC improves over methods that already exploit external KG information. BANER is a boundary-aware and LLM-assisted few-shot NER baseline that uses large language model knowledge or checking signals to improve candidate span verification [43]. These models correspond to the decomposed/span-based, knowledge-enhanced, and LLM-assisted categories in the related work. For the Few-NERD INTER and INTRA settings, we follow the comparison protocol reported by BANER and retain a compact set of representative baselines: ProtoBERT, CONTaiNER, DecomposedMetaNER, TadNER, KCL, KGen, and BANER [20,24,25,27–29,43]. For the industrial MaintIE and CFDK settings and the supplementary CoNLL-2003 setting, we additionally keep SpanProto when span-level comparison is available. All models are evaluated under the same data split, episode sampling protocol, metric, and evaluation script within each benchmark setting.

The source of the baseline numbers differs across benchmark groups. For MaintIE, CFDK, and CoNLL-2003, baseline models are evaluated under the same local episode construction and evaluation protocol used for KG-APC. For Few-NERD INTER and INTRA, the baseline scores are quoted from the BANER comparison setting because that work provides a recent and consistent few-shot NER comparison protocol. Accordingly, the Few-NERD tables are used as general-benchmark evidence, while the MaintIE and CFDK tables are the primary industrial evaluation. The inclusion of KCL and KGen makes the

comparison fairer for a KG-enhanced model, because KG-APC is now compared against both text-only few-shot NER models and knowledge-enhanced few-shot NER models. The ablation variant “w/o Gate” further serves as a controlled KG baseline: it uses the same Wikidata retrieval and KG encoder but replaces the K-aware adaptive gate with fixed fusion, thereby separating the effect of the adaptive correction mechanism from the effect of simply using external KG information.

4.3. Implementation Details

The default text encoder is RoBERTa-base [14]. For each input sentence, candidate spans are enumerated up to a maximum length of eight tokens, which covers most short maintenance mentions while avoiding excessive enumeration of long non-entity intervals. The KG encoder uses a two-layer graph attention network to aggregate type-neighborhood information [47]. The Wikidata-derived knowledge prior is aligned to the entity schema through type names, aliases, and descriptions, and the corrected prototype is then used for metric-based span classification. The default KG neighborhood is two hops, consistent with the hyperparameter analysis, and the shot-normalization constant in Equation (9) is set to $K_{\max} = 10$, matching the largest shot value considered in the industrial experiments.

We use AdamW for optimization, with a learning rate of 2×10^{-5} for the pretrained text encoder and 1×10^{-4} for newly introduced modules [48]. Each locally evaluated setting is run with five random seeds, and the average Micro-F1 score is reported. The boundary-aware loss weight λ_{BA} controls the contribution of the boundary-aware objective, and its influence is further examined in the hyperparameter analysis. For ablation experiments, only the target component is removed or replaced while the remaining training and evaluation settings are kept unchanged, so that the performance difference can be attributed to the removed component. During training, early stopping is based on validation Micro-F1, and the same stopping criterion is applied to KG-APC and locally evaluated baselines. The KG neighborhood retrieval is performed before episodic training and cached for each entity type, so the graph encoder adds limited per-episode overhead during inference.

4.3.1. Dataset Partitioning

For MaintIE and CFDK, we construct train, validation, and test splits at the text-instance level with a ratio of 60%/20%/20%. The splitting is stratified according to the entity-type distribution so that each split preserves the overall label distribution as much as possible. The same text instance is not shared across different splits. MaintIE is evaluated in two forms. In the coarse-grained setting, the five top-level classes, namely, PhysicalObject, State, Process, Activity, and Property, are used to construct 5-way episodes. In the fine-grained setting, episodes are sampled from the 32 intermediate fine-grained entity types. CFDK contains five target categories, namely, Equipment, Fault, Service Request, Cause, and Troubleshooting Solution, and is therefore evaluated under 5-way episodes.

For Few-NERD, we follow the official INTER and INTRA train/dev/test splits. The INTER split evaluates transfer across coarse domains, whereas the INTRA split evaluates fine-grained discrimination within related coarse domains. For CoNLL-2003, we use the standard train, development, and test partitions. Since CoNLL-2003 contains four entity types, namely, PER, LOC, ORG, and MISC, it is evaluated as a fixed 4-way few-shot setting with $K = 1$, $K = 5$, and $K = 10$.

4.3.2. Episode Construction

The locally evaluated experiments are conducted under an episodic evaluation protocol. For each N -way K -shot episode, we first sample N target entity types from the corresponding split. For each sampled type, K labeled entity mentions are selected to construct the support set. The query set is then constructed by sampling 10 query mentions

for each selected type. Support and query examples are sampled without overlap at the text-instance level. If a query text contains multiple target entities belonging to the sampled episode label space, all such entities are retained and evaluated. Entity types outside the current episode label space are treated as non-target types during episode-level evaluation. This description applies to our local evaluations; for Few-NERD baseline scores quoted from the BANER comparison protocol, we retain the published episodic settings described in the corresponding benchmark comparison.

For the industrial datasets, MaintIE coarse-grained and CFDK are evaluated under 5-way 1-shot, 5-way 5-shot, and 5-way 10-shot settings. MaintIE fine-grained is evaluated under 5-way 1-shot, 5-way 5-shot, 5-way 10-shot, 10-way 1-shot, 10-way 5-shot, and 10-way 10-shot settings. Few-NERD INTER and INTRA are reported under 5-way 1-shot, 5-way 5-shot, 10-way 1-shot, and 10-way 5-shot settings, following the general-benchmark comparison protocol. CoNLL-2003 is evaluated under 4-way fixed- K settings with $K = 1$, $K = 5$, and $K = 10$.

4.3.3. Validation Strategy and Number of Test Tasks

For locally evaluated settings, we sample 10,000 training episodes, 500 validation episodes, and 1000 test episodes for each random seed. The validation episodes are used only for early stopping and hyperparameter selection, while the test episodes are used only for final reporting. To reduce the effect of sampling randomness, the same sampled episodes are reused across locally evaluated models under the same setting and random seed. This statement does not apply to Few-NERD baseline scores quoted from the BANER comparison setting, where we follow the published baseline protocol. Each local experiment is repeated with five random seeds. The final result is reported as the average Micro-F1 over the five runs.

4.3.4. Early Stopping and Model Selection

During training, validation Micro-F1 is evaluated every 500 training episodes. The model checkpoint with the highest validation Micro-F1 is selected for final testing. Early stopping is applied when the validation Micro-F1 does not improve for five consecutive validation evaluations. The maximum number of training episodes is set to 10,000 for each setting. The same early stopping criterion is used for KG-APC and all locally evaluated baselines to ensure a fair comparison.

4.3.5. Implementation Environment

KG-APC and the locally evaluated baselines are implemented in Python 3.9 with PyTorch 2.1.0, CUDA 11.8, Transformers 4.36, PyTorch Geometric 2.4, NumPy 1.24, and scikit-learn 1.3. The experiments are conducted on a workstation equipped with two NVIDIA RTX 6000 Ada Generation GPUs with 48 GB memory each. Each individual run uses one GPU. Mixed-precision training is used to reduce GPU memory consumption, and the maximum gradient norm is set to 1.0.

4.3.6. Hyperparameter Selection

The default text encoder is RoBERTa-base. The maximum input length is set to 128 tokens for MaintIE, Few-NERD, and CoNLL-2003 and 256 tokens for CFDK because CFDK contains longer fault-diagnosis descriptions. Candidate spans are enumerated up to a maximum length of eight tokens. The default KG encoder is a two-layer graph attention network with four attention heads and dropout 0.1. The Wikidata neighborhood size is restricted to two hops in the default setting, and the retrieved KG neighbors are cached before episodic training. The shot-normalization constant is set to $K_{\max} = 10$.

AdamW is used for optimization. The learning rate is 2×10^{-5} for the pretrained RoBERTa encoder and 1×10^{-4} for newly introduced modules, including the span detector, graph encoder, gating module, and metric classifier. The weight decay is set to 0.01, and the dropout rate is set to 0.1. The episode batch size is 4 for 5-way settings and 2 for 10-way settings.

The main hyperparameters are selected according to validation Micro-F1. The KG neighborhood hop is searched in $\{1, 2, 3\}$, the boundary threshold τ_b is searched in $\{0.30, 0.40, 0.50, 0.60, 0.70\}$, and the boundary-aware loss weight λ_{BA} is searched in $\{0, 0.1, 0.3, 0.5, 0.7, 1.0\}$. Based on the validation results and the hyperparameter analysis, the default KG neighborhood is set to two hops. For the boundary threshold, $\tau_b = 0.60$ is used for MaintIE, $\tau_b = 0.40$ is used for CFDK, and $\tau_b = 0.50$ is used for Few-NERD and CoNLL-2003. For the boundary-aware loss weight, $\lambda_{BA} = 0.30$ is used for MaintIE and Few-NERD, $\lambda_{BA} = 0.50$ is used for CFDK, and $\lambda_{BA} = 0.20$ is used for CoNLL-2003.

4.4. Main Results

Tables 3–5 report the main results on the industrial maintenance datasets, which are the primary evaluation scenarios of this paper. Tables 6 and 7 provide additional results on Few-NERD as a general few-shot NER benchmark, and Table 8 gives a supplementary fixed-K comparison on CoNLL-2003. Therefore, the Few-NERD results are used to examine whether KG-APC remains competitive under a widely used general few-shot NER protocol, rather than to replace the industrial-dataset evidence. For Few-NERD, we use the corrected baseline scores reported under the same INTER and INTRA episode-level settings in BANER [43], which provides a stronger and more up-to-date comparison set for evaluating the proposed framework. The updated comparison also includes KCL and KGen as knowledge-enhanced baselines. This allows us to test whether the proposed adaptive prototype correction remains beneficial when the competing methods also use KG information.

In this paper, the summary phrase “the industrial settings” refers specifically to the 12 episodic configurations reported for MaintIE and CFDK in Tables 3–5: three MaintIE coarse-grained settings, six MaintIE fine-grained settings, and three CFDK settings. For each column, we subtract the strongest baseline Micro-F1 from the KG-APC Micro-F1 and then take the arithmetic mean of these 12 differences, yielding 1.89 percentage points. The Few-NERD and CoNLL-2003 results are general-domain supplementary evaluations and are not included in this industrial average.

Table 3. Main results on MaintIE coarse-grained few-shot NER (Micro-F1, %, mean $\pm$ std). All model results are locally evaluated under the same episodic protocol. The symbol $\dagger$ denotes a significant improvement over the strongest baseline in the same column (paired two-sided *t*-test, $p < 0.05$).

Model	5W-1S	5W-5S	5W-10S
ProtoBERT	55.84 $\pm$ 0.62	70.62 $\pm$ 0.48	75.84 $\pm$ 0.42
CONTaiNER	58.91 $\pm$ 0.58	73.48 $\pm$ 0.44	78.26 $\pm$ 0.39
DecomposedMetaNER	60.72 $\pm$ 0.55	75.15 $\pm$ 0.41	79.43 $\pm$ 0.37
SpanProto	62.38 $\pm$ 0.51	76.64 $\pm$ 0.39	80.72 $\pm$ 0.35
TadNER	64.21 $\pm$ 0.48	78.02 $\pm$ 0.36	82.06 $\pm$ 0.33
KCL	65.28 $\pm$ 0.46	78.21 $\pm$ 0.35	82.04 $\pm$ 0.32
KGen	66.18 $\pm$ 0.44	79.42 $\pm$ 0.34	83.06 $\pm$ 0.30
BANER	66.35 $\pm$ 0.52	79.28 $\pm$ 0.39	83.24 $\pm$ 0.35
KG-APC	68.02 $\pm$ 0.39 $\dagger$	81.05 $\pm$ 0.31 $\dagger$	85.13 $\pm$ 0.28 $\dagger$

Note: Bold values indicate the best result in each column.

All six comparative result tables report the mean and standard deviation of Micro-F1 over five random seeds. For the locally evaluated industrial settings, statistical signifi-

cance is assessed against the strongest baseline in the same column using a two-sided paired Student's *t*-test over the five matched-seed results. The symbol † denotes $p < 0.05$. Differences without this symbol are treated as descriptive and are not claimed to be statistically significant.

Table 3 reports the coarse-grained MaintIE results. Because this setting contains the five top-level MaintIE classes, the task mainly evaluates whether a model can separate broad maintenance concepts such as physical objects, states, processes, activities, and properties. KG-APC achieves the best Micro-F1 in all three settings, reaching 68.02 in 5-way 1-shot, 81.05 in 5-way 5-shot, and 85.13 in 5-way 10-shot. KCL and KGen improve over most text-only baselines, confirming that KG information is useful for coarse industrial entity categories. KGen is generally stronger than KCL because it uses KG-derived class features more directly for few-shot NER. Nevertheless, KG-APC still outperforms KGen by 1.84, 1.63, and 2.07 points in the three settings, and it improves over the strongest baseline in each column by 1.67, 1.63, and 1.89 points. All three improvements over the strongest baseline are statistically significant, and KG-APC has standard deviations of 0.28–0.39, indicating stable performance across the five seeds. The improvement remains stable when the number of support examples increases to 10, indicating that the KG prior is not only useful in the extremely low-shot case but also continues to provide complementary semantic information when more labeled examples are available. This result indicates that KG-APC benefits not merely from using a knowledge graph but from adaptively correcting prototypes after boundary-aware span detection.

Table 4. Main results on MaintIE fine-grained few-shot NER (Micro-F1, %, mean $\pm$ std). All model results are locally evaluated under the same episodic protocol. The symbol † denotes a significant improvement over the strongest baseline in the same column (paired two-sided *t*-test, $p < 0.05$).

Model	5W-1S	5W-5S	5W-10S	10W-1S	10W-5S	10W-10S
ProtoBERT	30.46 $\pm$ 0.86	43.82 $\pm$ 0.71	49.36 $\pm$ 0.62	23.85 $\pm$ 0.95	36.74 $\pm$ 0.78	42.28 $\pm$ 0.70
CONTaiNER	33.74 $\pm$ 0.82	47.96 $\pm$ 0.68	53.42 $\pm$ 0.58	26.52 $\pm$ 0.90	40.31 $\pm$ 0.74	45.86 $\pm$ 0.65
DecomposedMetaNER	35.81 $\pm$ 0.79	50.28 $\pm$ 0.64	55.61 $\pm$ 0.55	28.37 $\pm$ 0.86	42.46 $\pm$ 0.70	47.82 $\pm$ 0.62
SpanProto	38.26 $\pm$ 0.74	52.91 $\pm$ 0.59	58.34 $\pm$ 0.52	30.44 $\pm$ 0.82	44.87 $\pm$ 0.67	50.24 $\pm$ 0.58
TadNER	40.37 $\pm$ 0.71	55.12 $\pm$ 0.56	60.46 $\pm$ 0.50	32.18 $\pm$ 0.78	47.05 $\pm$ 0.63	52.38 $\pm$ 0.55
KCL	41.23 $\pm$ 0.68	56.38 $\pm$ 0.53	61.45 $\pm$ 0.48	33.11 $\pm$ 0.76	48.25 $\pm$ 0.60	53.64 $\pm$ 0.52
KGen	42.38 $\pm$ 0.65	57.51 $\pm$ 0.51	62.49 $\pm$ 0.46	34.10 $\pm$ 0.72	49.55 $\pm$ 0.57	54.73 $\pm$ 0.50
BANER	42.55 $\pm$ 0.70	57.38 $\pm$ 0.55	62.72 $\pm$ 0.49	34.26 $\pm$ 0.76	49.43 $\pm$ 0.60	54.91 $\pm$ 0.53
KG-APC	44.36 $\pm$ 0.58 †	59.24 $\pm$ 0.45 †	64.83 $\pm$ 0.41 †	36.08 $\pm$ 0.66 †	51.72 $\pm$ 0.51 †	57.16 $\pm$ 0.47 †

Note: Bold values indicate the best result in each column.

Table 4 shows the MaintIE fine-grained results. This setting is more difficult than the coarse-grained setting because episodes are sampled from 32 intermediate fine-grained entity types, and several types may share similar lexical forms or appear in short action-object expressions. KG-APC obtains the best result in all six settings. The comparison with KCL and KGen is particularly informative in this fine-grained setting. KCL improves the representation space through KG-guided contrastive learning, and KGen further strengthens label semantics through KG-derived class features; however, both still lag behind KG-APC. Compared with KGen, KG-APC improves Micro-F1 by 1.98, 1.73, and 2.34 points in the 5-way settings and by 1.98, 2.17, and 2.43 points in the 10-way settings. All six improvements over the strongest column-wise baselines are significant at $p < 0.05$. The standard deviations of KG-APC range from 0.41 to 0.66 and remain below those of the corresponding strongest baselines in these settings, supporting the stability of the improvement across seeds. The 10-shot results show that the performance advantage does not disappear when more support examples are provided. Instead, KG-APC still benefits from combining support-set evidence with structured type priors, especially when fine-grained entity types are semantically close. The larger gains in the fine-grained setting support the motivation

of adaptive prototype correction: when entity types are semantically close, using structured type priors helps stabilize class representations that would otherwise be constructed from very few support mentions, while boundary-aware span detection reduces the effect of partial or noisy industrial mentions.

Table 5. Main results on CFDK few-shot NER (Micro-F1, %, mean $\pm$ std). All model results are locally evaluated under the same episodic protocol. The symbol $\dagger$ denotes a significant improvement over the strongest baseline in the same column (paired two-sided t -test, $p < 0.05$).

Model	5W-1S	5W-5S	5W-10S
ProtoBERT	31.84 $\pm$ 0.91	44.72 $\pm$ 0.76	49.68 $\pm$ 0.68
CONTaiNER	34.26 $\pm$ 0.86	47.35 $\pm$ 0.72	52.11 $\pm$ 0.65
DecomposedMetaNER	35.58 $\pm$ 0.82	49.10 $\pm$ 0.69	53.67 $\pm$ 0.61
SpanProto	37.24 $\pm$ 0.78	51.08 $\pm$ 0.65	55.32 $\pm$ 0.58
TadNER	38.72 $\pm$ 0.74	52.46 $\pm$ 0.61	56.81 $\pm$ 0.55
KCL	39.03 $\pm$ 0.72	52.71 $\pm$ 0.59	57.02 $\pm$ 0.53
KGen	40.02 $\pm$ 0.70	53.86 $\pm$ 0.57	58.05 $\pm$ 0.51
BANER	40.15 $\pm$ 0.76	53.72 $\pm$ 0.62	58.23 $\pm$ 0.56
KG-APC	42.03 $\pm$ 0.63 $\dagger$	55.61 $\pm$ 0.50 $\dagger$	60.18 $\pm$ 0.46 $\dagger$

Note: Bold values indicate the best result in each column.

Table 5 reports the results on CFDK. Unlike MaintIE, CFDK focuses on compressor fault diagnosis log sheets and includes five domain-specific categories related to equipment, faults, service requests, causes, and troubleshooting solutions. Although the task is a five-way task, its practical difficulty is high because category complexity is uneven. Equipment spans are relatively easy to recognize, while Cause and Troubleshooting Solution frequently correspond to longer sequences that require accurate boundary detection and broader contextual understanding. This characteristic is consistent with the original CFDK study, where fully supervised models also achieved noticeably lower results on these difficult categories [5]. KG-APC obtains the best Micro-F1, with 42.03 in 5-way 1-shot, 55.61 in 5-way 5-shot, and 60.18 in 5-way 10-shot. KCL and KGen again outperform most purely text-based baselines, showing that knowledge-enhanced features are helpful for compressor fault diagnosis. However, KG-APC outperforms KGen by 2.01, 1.75, and 2.13 points, and it improves over the strongest baseline in each column by 1.88, 1.75, and 1.95 points. These three gains are statistically significant, while the standard deviations of 0.46–0.63 show that KG-APC remains stable despite the heterogeneous span lengths and category difficulty in CFDK. The 10-shot result shows that additional support examples improve all models, but KG-APC still maintains the strongest performance because graph-guided prototype correction provides information that is not fully captured by support examples alone. The consistent improvement on CFDK suggests that the proposed model is not limited to one maintenance annotation schema, but it can also benefit fault-diagnosis-oriented extraction where entity mentions are closely tied to equipment, causal descriptions, and troubleshooting procedures.

Table 6 presents the Few-NERD INTER results from the perspective of general few-shot NER evaluation. The INTER setting tests transfer to unseen entity types across coarse domains and therefore measure whether the proposed prototype correction mechanism can generalize beyond industrial maintenance schemas. KG-APC achieves the highest Micro-F1 in the 5-way 1-shot and 10-way 1-shot settings, with scores of 69.91 and 67.70, respectively. Compared with KGen, KG-APC improves by 2.33 points in 5-way 1-shot and 1.84 points in 10-way 1-shot, showing that adaptive prototype correction is especially useful when support evidence is extremely sparse. However, BANER outperforms KG-APC by 2.61 points in 5-way 5-shot and by 2.87 points in 10-way 5-shot. Thus, KG-APC does

not provide the best INTER result once five support examples are available. Because the Few-NERD baseline results follow the published BANER comparison protocol rather than a locally matched-seed rerun, these differences are interpreted descriptively and are not assigned significance markers. Thus, on this general benchmark, KG-APC is most competitive in extremely low-shot transfer settings, while LLM-assisted boundary verification remains stronger when more support examples are available.

Table 6. Results on Few-NERD INTER (Micro-F1, %, mean $\pm$ std). Baseline results are quoted from the BANER comparison protocol; KG-APC is evaluated under the same published episodic setting.

Model	5W-1S	5W-5S	10W-1S	10W-5S
ProtoBERT	38.83 $\pm$ 0.74	58.79 $\pm$ 0.62	32.45 $\pm$ 0.86	52.92 $\pm$ 0.70
CONTaiNER	50.92 $\pm$ 0.68	63.35 $\pm$ 0.58	47.02 $\pm$ 0.80	60.14 $\pm$ 0.66
DecomposedMetaNER	64.75 $\pm$ 0.55	71.49 $\pm$ 0.46	58.65 $\pm$ 0.71	68.11 $\pm$ 0.54
TadNER	64.83 $\pm$ 0.52	72.12 $\pm$ 0.43	64.06 $\pm$ 0.66	69.94 $\pm$ 0.51
KCL	65.84 $\pm$ 0.50	71.26 $\pm$ 0.44	63.57 $\pm$ 0.65	67.43 $\pm$ 0.53
KGen	67.58 $\pm$ 0.46	73.18 $\pm$ 0.40	65.86 $\pm$ 0.60	69.05 $\pm$ 0.49
BANER	69.26 $\pm$ 0.44	76.53 $\pm$ 0.37	67.43 $\pm$ 0.55	72.24 $\pm$ 0.45
KG-APC	69.91 $\pm$ 0.42	73.92 $\pm$ 0.39	67.70 $\pm$ 0.53	69.37 $\pm$ 0.48

Note: Bold values indicate the best result in each column.

Table 7. Results on Few-NERD INTRA (Micro-F1, %, mean $\pm$ std). Baseline results are quoted from the BANER comparison protocol; KG-APC is evaluated under the same published episodic setting.

Model	5W-1S	5W-5S	10W-1S	10W-5S
ProtoBERT	20.76 $\pm$ 0.82	42.54 $\pm$ 0.71	15.05 $\pm$ 0.94	35.40 $\pm$ 0.79
CONTaiNER	41.51 $\pm$ 0.72	57.83 $\pm$ 0.60	36.62 $\pm$ 0.84	51.04 $\pm$ 0.70
DecomposedMetaNER	49.48 $\pm$ 0.66	62.92 $\pm$ 0.54	42.84 $\pm$ 0.78	57.31 $\pm$ 0.63
TadNER	60.78 $\pm$ 0.52	67.94 $\pm$ 0.46	55.44 $\pm$ 0.68	60.87 $\pm$ 0.57
KCL	61.42 $\pm$ 0.51	67.96 $\pm$ 0.45	56.38 $\pm$ 0.66	60.72 $\pm$ 0.58
KGen	63.28 $\pm$ 0.47	69.35 $\pm$ 0.42	58.16 $\pm$ 0.61	62.31 $\pm$ 0.54
BANER	64.95 $\pm$ 0.43	72.14 $\pm$ 0.38	61.24 $\pm$ 0.55	67.53 $\pm$ 0.49
KG-APC	65.02 $\pm$ 0.45	70.51 $\pm$ 0.41	59.11 $\pm$ 0.59	62.96 $\pm$ 0.53

Note: Bold values indicate the best result in each column.

Table 7 reports the Few-NERD INTRA results as another general few-shot NER comparison. In this setting, fine-grained entity types are sampled within related coarse domains, so the task emphasizes discrimination among semantically similar types. KG-APC obtains the best score in the 5-way 1-shot setting, slightly outperforming BANER. It also consistently outperforms KCL and KGen in all INTRA settings, with gains over KGen of 1.74, 1.16, 0.95, and 0.65 points. Nevertheless, the 0.07-point difference between KG-APC and BANER in 5-way 1-shot is small relative to their reported standard deviations and is therefore only a numerical advantage, not a statistically significant one. BANER exceeds KG-APC by 1.63 points in 5-way 5-shot, 2.13 points in 10-way 1-shot, and 4.57 points in 10-way 5-shot. This pattern shows that KG-APC can provide useful prototype correction under the most data-scarce general NER setting, but its advantage becomes smaller when additional support examples make text-only prototypes more reliable and when LLM-assisted boundary checking is available. The result also clarifies the role of Few-NERD in this paper: it serves as a general benchmark for testing transferability and competitiveness, whereas the main experimental evidence for the target industrial scenario comes from MaintIE and CFDK.

Table 8 shows the fixed- K CoNLL-2003 results. CoNLL-2003 has a smaller and coarser label space than Few-NERD and the industrial fine-grained settings, so all strong baselines obtain relatively high scores. In this supplementary general-domain benchmark, BANER obtains the highest Micro-F1 in every setting and exceeds KG-APC by 1.05, 0.77, and

0.27 points for $K = 1$, $K = 5$, and $K = 10$, respectively. KG-APC outperforms KCL and KGen in all three settings, but this does not alter the fact that it is not the strongest method on CoNLL-2003. No significance claim is therefore made. The smaller benefit of prototype correction is consistent with CoNLL-2003's four broad news-domain categories, for which fine-grained type-level KG priors offer less additional information than in MaintIE or CFDK.

Table 8. Results on CoNLL-2003 fixed- K few-shot NER (4-way, Micro-F1, %, mean $\pm$ std). All model results are locally evaluated under the same fixed- K protocol.

Model	$K = 1$	$K = 5$	$K = 10$
ProtoBERT	70.82 $\pm$ 0.45	80.94 $\pm$ 0.36	84.17 $\pm$ 0.31
CONTaiNER	73.41 $\pm$ 0.42	83.05 $\pm$ 0.33	85.66 $\pm$ 0.28
DecomposedMetaNER	74.68 $\pm$ 0.40	83.87 $\pm$ 0.32	86.20 $\pm$ 0.27
SpanProto	76.83 $\pm$ 0.37	85.50 $\pm$ 0.30	87.51 $\pm$ 0.25
TadNER	78.04 $\pm$ 0.35	86.32 $\pm$ 0.28	88.29 $\pm$ 0.24
KCL	76.94 $\pm$ 0.38	85.28 $\pm$ 0.30	87.62 $\pm$ 0.26
KGen	77.86 $\pm$ 0.36	85.91 $\pm$ 0.28	88.04 $\pm$ 0.24
BANER	79.70 $\pm$ 0.31	86.94 $\pm$ 0.25	88.65 $\pm$ 0.21
KG-APC	78.65 $\pm$ 0.34	86.17 $\pm$ 0.27	88.38 $\pm$ 0.23

Note: Bold values indicate the best result in each column.

4.5. Computational Efficiency Comparison

The complexity analysis in Section 3.5 indicates that KG-APC avoids the online cost of repeatedly retrieving Wikidata neighborhoods and does not require an LLM-scale backbone. We therefore conduct a computational efficiency comparison after the main recognition results. All latency, throughput, FLOPs, and peak-memory measurements are obtained under FP16 inference on a single NVIDIA RTX 6000 Ada Generation GPU. Latency and FPS are measured at the query-text level, and the same input-length setting is used for KG-APC and BANER within each dataset.

Figure 2 shows a consistent time-efficiency advantage over BANER on all three industrial settings. KG-APC reduces per-query latency by factors of 54.5, 54.2, and 40.6 on MaintIE coarse-grained, MaintIE fine-grained, and CFDK, respectively, while increasing throughput by corresponding factors of 54.6, 54.1, and 40.4. The advantage remains visible after accounting for different sequence lengths: KG-APC requires 26.8–62.4 G FLOPs per query, whereas BANER requires 0.98–1.72 T FLOPs, corresponding to a 27.6–36.6-fold reduction. Peak inference memory is also reduced from 15.6–17.8 GB to 2.0–2.6 GB, a 6.8–7.8-fold reduction across the three datasets. CFDK remains slower for both models because its fault-diagnosis descriptions are longer, but KG-APC retains a substantial advantage under this more demanding input condition.

Figure 3 explains the space and training-time differences. KG-APC loads 132 M parameters, compared with 7.0B for BANER, yielding 53 times fewer resident parameters after rounding to the nearest whole-number factor. Its 2.2 GB peak inference memory is 6.8–8.2 times lower than the 15–18 GB required by BANER, and its 2.6 h training time per seed is 4.6–6.9 times shorter than BANER's 12–18 h range. The graph encoder and adaptive correction modules introduce only a modest overhead relative to the RoBERTa-based SpanProto, TadNER, and KGen baselines: KG-APC adds 2–6 M parameters, 0.1–0.5 GB of peak memory, and 0.2–0.8 h of training time. Thus, the proposed method does not claim to be the smallest encoder-based baseline; rather, it obtains its knowledge-enhanced and boundary-aware gains while remaining close to compact RoBERTa-based methods and substantially below the time and space requirements of the LLM-assisted BANER pipeline.

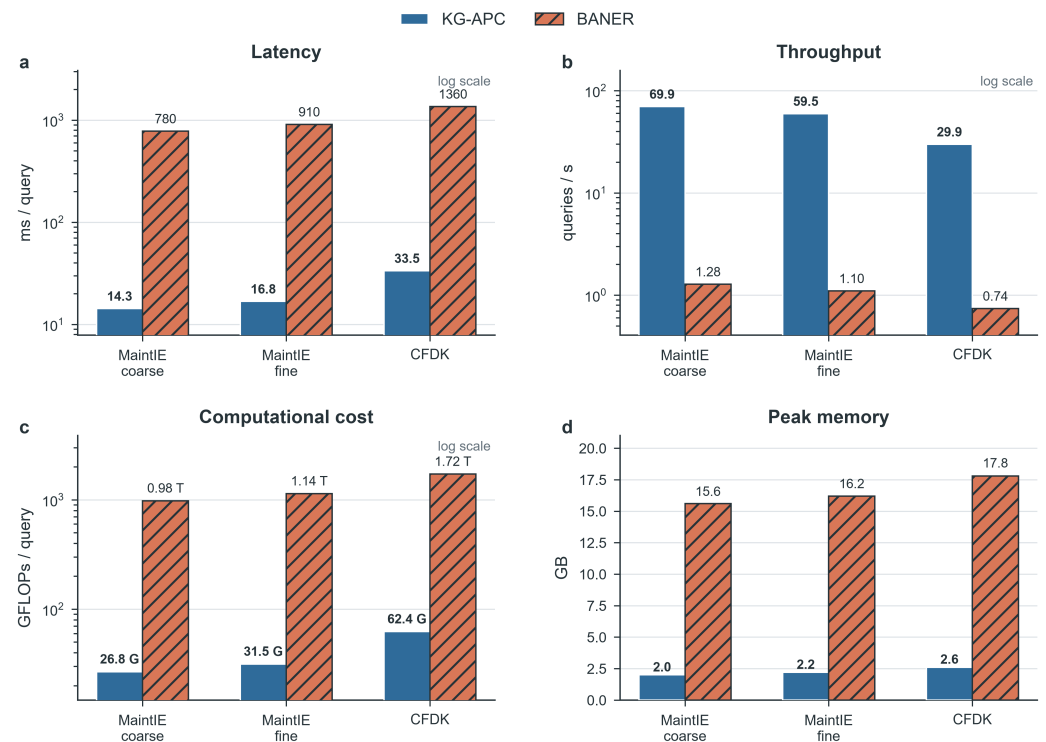


Figure 2. Visual comparison of FP16 inference efficiency between KG-APC and BANER on the three industrial NER settings: (a) latency per query, (b) throughput in queries per second, (c) FLOPs per query, and (d) peak inference memory. Panels (a–c) use logarithmic vertical axes to preserve the visibility of both methods across their large magnitude differences; panel (d) uses a linear vertical axis. Lower values indicate better efficiency except in panel (b), where higher FPS is better.

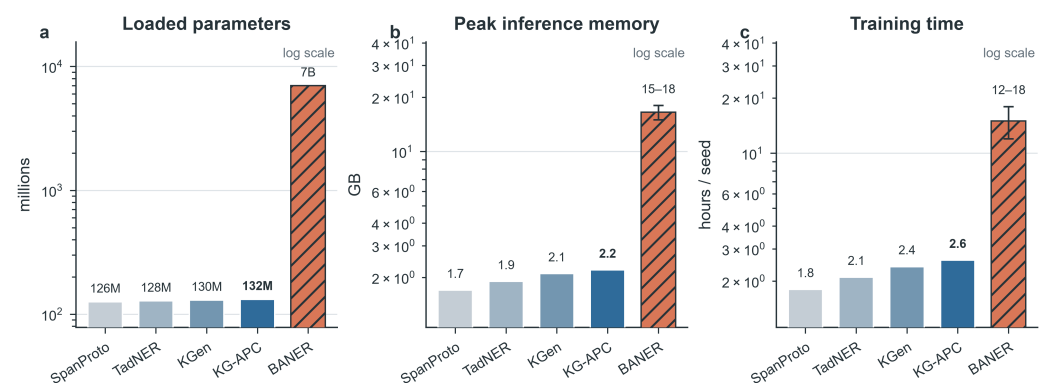


Figure 3. Visual comparison of model and training costs: (a) loaded parameters, (b) peak inference memory, and (c) training time per seed. All panels use logarithmic vertical axes because BANER operates at a substantially larger scale than the RoBERTa-based methods. The error bars and range labels for BANER in panels (b,c) represent the measured hardware-dependent ranges, rather than statistical variability across seeds.

Table 9 further shows that the external-knowledge overhead is small and occurs offline. Even for MaintIE fine-grained NER, which contains 32 entity types, retrieving the two-hop neighborhoods takes 22.7 s and produces a 17.9 MB cache. The other datasets require 2.4–3.9 s and 2.1–3.0 MB. Because these neighborhoods and type priors are reused across episodes, this one-time cost does not add online Wikidata queries to per-query inference. Taken together, the two efficiency figures and the Wikidata-cost table support the practical efficiency claim in a bounded form: under the reported RTX 6000 Ada FP16 setting, KG-

APC is markedly more efficient than BANER in both time and space while retaining the recognition advantages reported on the industrial datasets.

Table 9. Cost of retrieving and caching Wikidata neighborhoods for KG-APC. Retrieval is performed once before episodic training and is not repeated during online inference.

Dataset	Entity Types	Avg. Nodes/Type	Avg. Edges/Type	Retrieval Time	Cache Size
MaintIE coarse	5	48	136	3.6 s	2.8 MB
MaintIE fine	32	46	131	22.7 s	17.9 MB
CFDK	5	50	142	3.9 s	3.0 MB
CoNLL-2003	4	42	118	2.4 s	2.1 MB

4.6. Ablation Study

We conduct ablation experiments on the industrial few-shot NER settings rather than on Few-NERD because the main motivation of KG-APC is to improve low-resource entity extraction in maintenance information systems. Tables 10 and 11 report the results on MaintIE fine-grained and CFDK under the 5-way 5-shot setting, respectively. The full KG-APC rows are aligned with the corresponding 5-way 5-shot KG-APC results in Tables 4 and 5. We evaluate three variants: removing the KG prior and using only text prototypes, replacing the adaptive gate with fixed fusion, and removing boundary-aware span detection. This design tests whether the improvements come from structured prior knowledge, dynamic prototype correction, and explicit span modeling. The “w/o Gate” variant is also the controlled KG-prior baseline: it retains the same Wikidata alignments, two-hop neighborhoods, graph encoder, textual prototypes, and episodic data as KG-APC, but it replaces the K -aware adaptive gate with fixed prototype–prior fusion. It therefore isolates whether the adaptive correction mechanism adds value beyond access to the same external Wikidata signal.

Table 10. Ablation results on MaintIE fine-grained 5-way 5-shot.

Variant	Prec.	Rec.	F1	$\Delta F1$
w/o KG	58.04	56.23	57.12	−2.12
w/o Gate	58.61	57.17	57.88	−1.36
w/o BA-SD	56.86	58.15	57.50	−1.74
KG-APC Full	60.18	58.33	59.24	–

Note: Bold values indicate the best result in each column.

Table 10 shows that all three components contribute to MaintIE fine-grained recognition. Removing the KG prior causes the largest F1 drop, from 59.24 to 57.12, indicating that structured type information is important when episodes are sampled from semantically close fine-grained maintenance categories. Under the controlled same-KG comparison, replacing the adaptive gate with fixed fusion reduces F1 from 59.24 to 57.88, a decrease of 1.36 points. Removing BA-SD decreases F1 to 57.50, confirming that boundary quality remains important even after the model obtains corrected class prototypes.

Table 11. Ablation results on CFDK 5-way 5-shot.

Variant	Prec.	Rec.	F1	$\Delta F1$
w/o KG	54.10	52.41	53.24	−2.37
w/o Gate	54.86	53.42	54.13	−1.48
w/o BA-SD	52.98	54.63	53.79	−1.82
KG-APC Full	56.48	54.77	55.61	–

Note: Bold values indicate the best result in each column.

Table 11 shows a similar trend on CFDK. The full model reaches 55.61 F1, while the w/o KG, w/o Gate, and w/o BA-SD variants decrease to 53.24, 54.13, and 53.79, respectively. The largest degradation again comes from removing the KG prior, which is consistent with the fault-diagnosis nature of CFDK: equipment, fault, cause, and solution mentions are semantically connected, and graph-based priors help represent these dependencies under few-shot supervision. With the same Wikidata signal retained, fixed fusion in the w/o Gate variant is 1.48 F1 points below KG-APC, corroborating the contribution of adaptive prototype correction. The BA-SD variant also shows a clear drop. This is important for CFDK because the difficult Cause and Troubleshooting Solution categories often appear as long spans; if their boundaries are truncated or mixed with neighboring fault descriptions, the subsequent prototype comparison becomes unreliable.

4.7. Robustness to Incomplete and Noisy Knowledge Graphs

The usefulness of a graph-derived prior depends on both KG correctness and KG coverage. Large knowledge graphs inevitably balance completeness against correctness, and automatically retrieved neighborhoods may contain irrelevant relations, missing concepts, or weak entity alignments [1,2,49]. These issues are particularly relevant in industrial maintenance, where equipment taxonomies and fault vocabularies evolve over time. We therefore evaluate two complementary failure modes: semantic noise in otherwise available two-hop neighborhoods and missing Wikidata alignments for episode-level entity types. Both experiments use the MaintIE fine-grained and CFDK 5-way 5-shot settings, the same test episodes and Micro-F1 metric as the main experiments, and the five-seed local evaluation protocol described in Section 4.3. Only the KG input is perturbed; the text data, episode construction, and non-KG hyperparameters remain unchanged. The complete perturbation and reproducibility procedures are provided in Supplementary Note S1; the main text retains the essential experimental definitions and the corresponding results.

4.7.1. Scale of the Constructed KG Priors

The robustness tests operate on cached, relation-filtered two-hop Wikidata priors [3]. Table 9 reports their dataset-level scale and caching cost; for the largest industrial schema, MaintIE fine-grained NER, each type has 46 nodes and 131 edges on average, and the complete cache occupies 17.9 MB.

4.7.2. Robustness to Noisy KG Neighborhoods

We test semantic contamination by independently replacing retained two-hop neighbors with unrelated Wikidata nodes at probabilities $r \in \{0, 0.1, 0.2, 0.3, 0.5\}$. Neighborhood size and feature dimensionality remain fixed, so the comparison isolates semantic noise rather than graph-scale variation. This perturbation represents irrelevant concepts introduced by automatic neighborhood expansion or incorrect relation retrieval [50].

Table 12 shows monotonic but gradual degradation as the replacement probability increases. At a replacement probability of 10%, Micro-F1 decreases by 0.32 points on MaintIE fine-grained NER and 0.39 points on CFDK; at 20%, the corresponding reductions remain below one point. At a replacement probability of 50%, KG-APC reaches 57.38 and 53.69 F1, remaining 0.26 and 0.45 points above the respective w/o KG variants. The monotonic decline confirms that neighborhood quality matters, whereas the residual advantage under the 50% condition indicates that partially correct graph evidence can remain useful. This pattern is consistent with the adaptive interpolation retaining a support-set prototype path instead of replacing textual evidence with the KG prior. Because the experiment does not isolate the gate under each replacement level, however, the observed robustness should not be attributed to the gate alone; relation filtering and the remaining clean neighbors may also contribute.

Table 12. Sensitivity to noisy KG neighborhood injection on MaintIE fine-grained NER and CFDK. Values are the mean Micro-F1 (%) over five seeds under the 5-way 5-shot setting; $\Delta F1$ is measured relative to the clean 0% condition.

Replacement Probability	MaintIE Fine F1	$\Delta F1$	CFDK F1	$\Delta F1$
0%	59.24	–	55.61	–
10%	58.92	−0.32	55.22	−0.39
20%	58.46	−0.78	54.76	−0.85
30%	58.03	−1.21	54.31	−1.30
50%	57.38	−1.86	53.69	−1.92
w/o KG	57.12	−2.12	53.24	−2.37

Note: Bold values indicate the best result in each F1 column.

4.7.3. Robustness to Missing Entity-Type Alignments

We evaluate incomplete KG coverage by independently masking episode-level type alignments at probabilities $m \in \{0, 0.25, 0.5, 0.75, 1\}$. A masked type bypasses the KG branch and falls back to its support-set textual prototype. This setting represents newly introduced equipment, fault modes, or maintenance operations for which no reliable Wikidata anchor is available.

Table 13 also exhibits controlled monotonic degradation. At a masking probability of 25%, KG-APC loses 0.53 F1 on MaintIE fine-grained NER and 0.59 F1 on CFDK. At a masking probability of 50%, the model still achieves 58.05 and 54.42 F1, respectively. Masking all type alignments is equivalent to bypassing the KG branch for every type and exactly matches the w/o KG ablation, providing an internal consistency check on the fallback implementation. The results therefore support two bounded conclusions: KG coverage is beneficial because performance declines as more alignments are removed, and the model does not fail abruptly when coverage is incomplete because the support-set prototype remains available.

Table 13. Sensitivity to missing KG type alignments on MaintIE fine-grained NER and CFDK. Values are the mean Micro-F1 (%) over five seeds under the 5-way 5-shot setting; $\Delta F1$ is measured relative to the fully aligned 0% condition.

Masking Probability	MaintIE Fine F1	$\Delta F1$	CFDK F1	$\Delta F1$
0%	59.24	–	55.61	–
25%	58.71	−0.53	55.02	−0.59
50%	58.05	−1.19	54.42	−1.19
75%	57.54	−1.70	53.83	−1.78
100%	57.12	−2.12	53.24	−2.37

Note: Bold values indicate the best result in each F1 column.

Table 14 extends the w/o Gate ablation in Tables 10 and 11 to incomplete KG coverage. KG-APC Full and w/o Gate use identical test episodes, masked type assignments, KG neighborhoods, textual prototypes, and non-gate settings within each random seed; as defined in the ablation study, w/o Gate replaces the adaptive gate with fixed prototype–prior fusion. At 0% masking, KG-APC Full is numerically higher than w/o Gate by 1.36 F1 points on MaintIE fine-grained NER and 1.48 points on CFDK, reproducing the clean-condition differences in the ablation study. At the representative 50% masking probability, KG-APC Full obtains 58.05 and 54.42 F1, compared with 57.36 and 53.78 for w/o Gate, corresponding to differences of 0.69 and 0.64 points. The smaller difference at 50% masking is consistent with fewer aligned types entering prototype–prior fusion because masked types use the textual-prototype fallback. Thus, the controlled comparison supports a contribution of the adaptive gate under this representative incomplete-coverage condition.

Table 14. Comparison of KG-APC Full and w/o Gate under 0% and 50% KG type-alignment masking. Values are the mean Micro-F1 (%) over five seeds in the 5-way 5-shot setting.

Masking Probability	Variant	MaintIE Fine F1	CFDK F1
0%	KG-APC Full	59.24	55.61
0%	w/o Gate	57.88	54.13
50%	KG-APC Full	58.05	54.42
50%	w/o Gate	57.36	53.78

Note: Bold values indicate the better result for each masking probability and dataset.

Taken together, these experiments show that KG-APC remains functional under random neighborhood contamination and partial type coverage, while also confirming that cleaner and more complete priors produce better recognition. The tests represent static perturbations of cached neighborhoods; they do not reproduce temporally correlated graph drift, systematic misalignment of an entire type family, or adversarial relation changes. Accordingly, these results demonstrate robustness under the evaluated noise and missing-coverage conditions. In deployment, cached priors could be refreshed when new concepts become available, while explicit KG-confidence assessment and noise-aware relation filtering remain important directions for future work.

4.8. Case Study

To complement the aggregate results with a traceable example, we analyze the publicly available MaintIE query “change out engine” under a 5-way 5-shot fine-grained episode [4]. Its gold annotations contain the complete action span “change out”, labeled as Activity/MaintenanceActivity, and the equipment span “engine”, labeled as PhysicalObject/DrivingObject. The compact action–object construction is useful for illustrating both error sources targeted by KG-APC: a boundary model can retain an incomplete or over-extended interval, while a prototype classifier can confuse semantically close fine-grained types. This single case is intended to expose the decision process; the dataset-level effectiveness claims remain supported by the main and ablation results rather than by the example alone.

The episode label set additionally contains Activity/SupportingActivity, PhysicalObject/GeneratingObject, and State/UndesirableState. With only five support examples per class, the textual prototypes may not cover every phrasal maintenance action or equipment alias. We therefore compare two otherwise identical classification paths. The text-only path uses the initial support-set prototype $\tilde{p}_c$, following standard prototype-based classification [22]; the corrected path uses p_c , which fuses the textual prototype with the Wikidata-derived type prior [3]. Table 15 reports the resulting candidate-level decisions. Values in the BA-SD column are boundary confidence scores B_{ij} , whereas values in the two prototype columns are the post-softmax probabilities of the displayed predicted classes.

The boundary decisions in Table 15 are consistent with the MaintIE threshold $\tau_b = 0.60$ specified in Section 4.3. The incomplete span “change” and the mixed or over-extended spans “out engine” and “change out engine” obtain scores of 0.31, 0.34, and 0.42 and are therefore rejected. In contrast, “change out” and “engine” obtain scores of 0.84 and 0.91 and proceed to type classification. The example makes explicit how span-level filtering prevents malformed semantic units from entering prototype matching, consistent with the motivation of decomposed and span-based few-shot NER [26,27].

Table 15. Case study of boundary-aware span detection and KG-guided prototype correction on a MaintIE query. Text-only and corrected decisions use $\tilde{p}_c$ and p_c , respectively.

Candidate Span	BA-SD Decision	Text-Only Prototype Decision	KG-Corrected Prototype Decision	Gold Label
change	Reject (0.31)	Not evaluated	Not evaluated	Non-entity
change out	Retain (0.84)	Activity / SupportingActivity (0.61)	Activity / MaintenanceActivity (0.76)	Activity / MaintenanceActivity
engine	Retain (0.91)	PhysicalObject / GeneratingObject (0.58)	PhysicalObject / DrivingObject (0.74)	PhysicalObject / DrivingObject
out engine	Reject (0.34)	Not evaluated	Not evaluated	Non-entity
change out engine	Reject (0.42)	Not evaluated	Not evaluated	Non-entity

Prototype correction then changes both retained-span decisions. For “change out”, the text-only prototype predicts Activity/SupportingActivity with probability 0.61, whereas the KG-corrected prototype predicts the gold type Activity/MaintenanceActivity with probability 0.76. The relevant retained Wikidata neighborhood includes repair, replacement, inspection, and calibration, which provides a type-level semantic anchor for the maintenance action. For “engine”, the text-only path predicts PhysicalObject/GeneratingObject with probability 0.58. After correction with neighbors related to the motor, engine, actuator, and drive unit, the prediction changes to the gold PhysicalObject/DrivingObject type with probability 0.74.

The case therefore illustrates the intended division of labor in KG-APC: BA-SD removes incomplete and over-extended intervals before typing, while adaptive prototype correction resolves ambiguity among retained fine-grained spans. It also reveals the boundary of this evidence. The example shows how a successful correction occurs, but it does not establish how frequently each error mode appears; those claims are instead evaluated by the dataset-level comparisons and component ablations.

4.9. Hyperparameter Analysis

To further investigate the robustness of KG-APC, we conduct a series of hyperparameter analyses on MaintIE fine-grained NER and CFDK. Specifically, we analyze the effects of the number of support examples, the neighborhood hop of the knowledge graph encoder, the boundary threshold in BA-SD, and the weight of the boundary-aware loss. Unless otherwise specified, the hyperparameter experiments are conducted under the 5-way 5-shot setting, and Micro-F1 is used as the evaluation metric.

4.9.1. K-Shot Sensitivity Analysis

Figure 4 reports the effect of different numbers of support examples on MaintIE fine-grained NER and CFDK. As the number of support examples increases from 1 to 5, the performance of KG-APC improves substantially on both datasets. On MaintIE fine-grained NER, the Micro-F1 score increases from 44.36% to 59.24%, while on CFDK it increases from 42.03% to 55.61%. This demonstrates that additional support instances can effectively alleviate the instability of class prototypes under extremely low-resource conditions.

When the number of support examples further increases from 5 to 10, the performance continues to improve, but the gain becomes smaller. Specifically, KG-APC achieves 64.83% on MaintIE fine-grained NER and 60.18% on CFDK under the 10-shot setting. The marginal improvement from 5-shot to 10-shot is lower than that from 1-shot to 5-shot, indicating that prototype construction gradually becomes saturated as more support samples are provided. Overall, the results confirm that KG-APC benefits from additional support examples while maintaining stable performance across different few-shot settings.

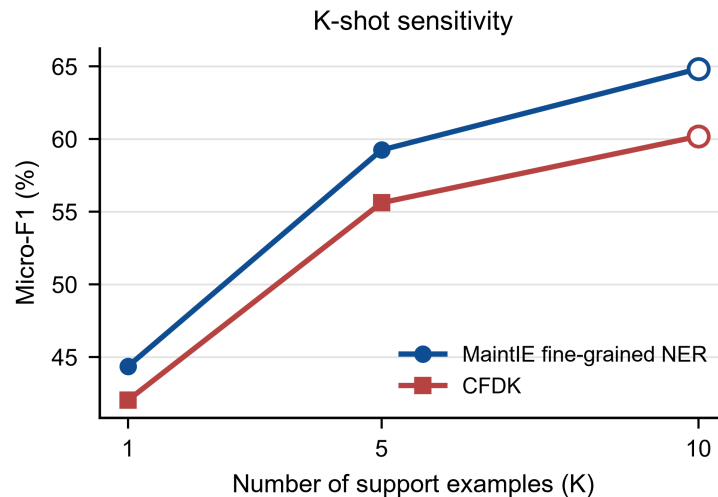


Figure 4. K-shot sensitivity on MaintIE fine-grained NER and CFDK. The curves show that increasing the number of support examples improves Micro-F1 on both datasets, while the marginal gain decreases from 5-shot to 10-shot.

4.9.2. Effect of Neighborhood Hop in the Knowledge Graph Encoder

Figure 5 shows the influence of the neighborhood hop in the knowledge graph encoder. In our model, the neighborhood hop corresponds to the number of GAT layers used to aggregate structural information from the knowledge graph. A 1-hop setting only incorporates direct neighbors, whereas a larger hop allows the model to capture higher-order semantic relations among entity types.

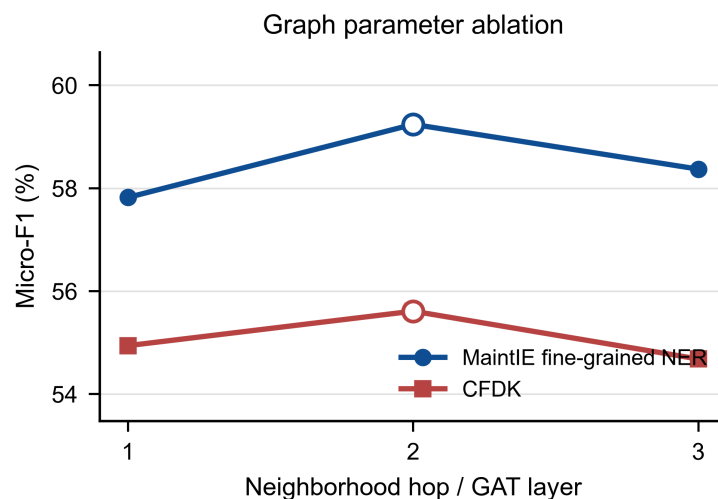


Figure 5. Graph parameter ablation with different neighborhood hops. Both datasets reach the best Micro-F1 with a 2-hop knowledge graph neighborhood, whereas 3-hop aggregation introduces weaker or noisier neighbors and reduces performance.

The best performance is obtained when the neighborhood hop is set to 2 on both datasets. On MaintIE fine-grained NER, the Micro-F1 score increases from 57.82% with one-hop aggregation to 59.24% with two-hop aggregation. Similarly, on CFDK, the performance improves from 54.94% to 55.61%. These results suggest that second-order neighborhood information provides useful semantic priors for prototype correction. However, when the hop is further increased to 3, the performance drops to 58.37% on MaintIE and 54.68% on CFDK. This degradation may be caused by the introduction of noisy or weakly related neighbors, which can lead to over-smoothing of type representations and reduce the

discriminability of corrected prototypes. Therefore, a two-hop neighborhood is adopted as the default setting.

4.9.3. Effect of Boundary Threshold in BA-SD

Figure 6 presents the effect of the boundary threshold τ_b in the boundary-aware span detector. The threshold τ_b controls whether a candidate span is retained according to its boundary confidence score. A smaller threshold tends to preserve more candidate spans and improve recall, whereas a larger threshold imposes stricter filtering and may improve precision.

The optimal value of τ_b differs across the two datasets. On MaintIE fine-grained NER, the best Micro-F1 score of 59.24% is achieved when $\tau_b = 0.60$. MaintIE contains many relatively short and fine-grained industrial entity mentions, and a stricter threshold helps remove noisy candidate spans before type classification. In contrast, CFDK achieves the best performance when $\tau_b = 0.40$, with a Micro-F1 score of 55.61%. This is because CFDK contains more long-span expressions, especially for cause and troubleshooting solution labels. If the threshold is too high, long spans are more likely to be incorrectly filtered out, leading to a decrease in recall. Therefore, CFDK requires a relatively lower boundary threshold to retain sufficient long-span candidates.

These results indicate that the optimal boundary threshold is closely related to dataset-specific span characteristics. For datasets with shorter and more regular entity mentions, a stricter threshold can improve span precision. For datasets with longer and more heterogeneous spans, a lower threshold is more suitable for preserving recall.

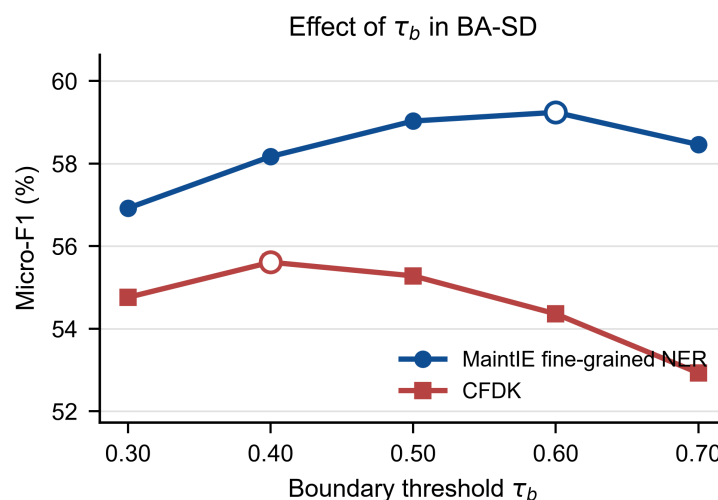


Figure 6. Effect of the boundary threshold τ_b in BA-SD. MaintIE fine-grained NER favors a stricter threshold, while CFDK performs best with a lower threshold because its Cause and Troubleshooting Solution entities often form longer spans.

4.9.4. Effect of Boundary-Aware Loss Weight

Figure 7 reports the effect of the boundary-aware loss weight λ_{BA} . This hyperparameter controls the contribution of the boundary-aware objective to the overall training loss. When $\lambda_{BA} = 0$, the boundary-aware loss is removed, and the model relies mainly on prototype-based type classification. In this case, the performance drops to 56.42% on MaintIE fine-grained NER and 52.86% on CFDK, demonstrating the importance of explicit boundary-aware learning.

The optimal value of λ_{BA} is different for the two datasets. On MaintIE fine-grained NER, the best result is obtained when $\lambda_{BA} = 0.3$, reaching 59.24% Micro-F1. A moderate boundary-aware loss can improve span representation quality without overwhelming

the type classification objective. However, when λ_{BA} becomes too large, the model may overemphasize boundary discrimination and weaken fine-grained type representation learning, resulting in performance degradation.

On CFDK, the best performance is achieved when $\lambda_{BA} = 0.5$, with a Micro-F1 score of 55.61%. Compared with MaintIE, CFDK contains more long and structurally diverse spans, making boundary modeling more important for accurate entity recognition. Therefore, CFDK benefits from a larger boundary-aware loss weight. However, further increasing λ_{BA} to 0.7 also reduces performance, indicating that excessive boundary supervision may still interfere with prototype learning and semantic classification.

Overall, the results show that λ_{BA} should not be fixed uniformly across datasets. MaintIE fine-grained NER benefits from a moderate boundary-aware constraint, while CFDK requires a stronger boundary-aware signal due to its longer and more heterogeneous span structures.

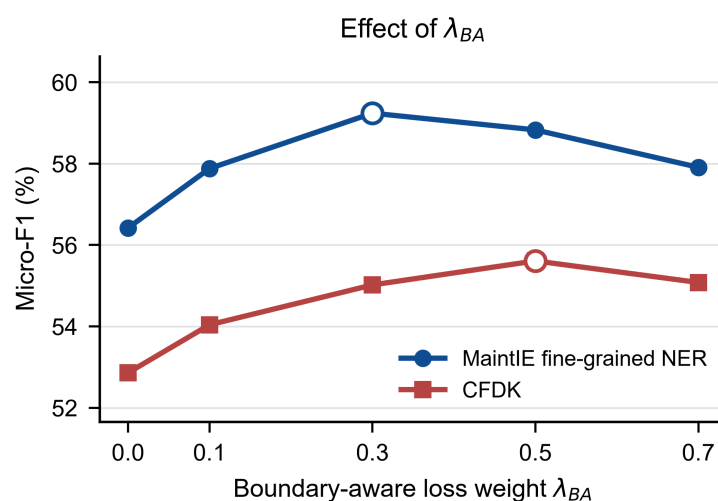


Figure 7. Effect of the boundary-aware loss weight λ_{BA} . MaintIE fine-grained NER obtains the best result with a moderate boundary-aware constraint, whereas CFDK benefits from a stronger boundary signal due to its longer and more heterogeneous span structures.

5. Discussion

The main finding of this study is that few-shot industrial NER benefits from treating boundary detection and type representation as coupled problems rather than as two independent sources of error. In industrial maintenance texts, an entity mention is often short, domain-specific, and embedded in compact action-object or fault-description expressions. If the candidate span is incomplete, the following type classifier receives a distorted semantic unit; if the class prototype is constructed only from one or a few support mentions, the classifier can be biased by accidental lexical or contextual features. KG-APC addresses these two problems by combining boundary-aware span detection with adaptive prototype correction. The consistent improvements on MaintIE and CFDK indicate that the framework is especially useful when span boundaries are ambiguous and type semantics are strongly connected to external domain knowledge.

The results also clarify how KG-APC relates to existing few-shot NER research. Prototype-based and metric-learning methods provide an efficient foundation for low-resource recognition, but their class representations can become unstable when the support set is extremely small [20,22]. Decomposed and span-based methods reduce this difficulty by separating span detection from type classification, which is helpful for NER tasks where boundary errors dominate final performance [25–27]. KG-APC follows this decomposed philosophy but further introduces a Wikidata-derived graph prior to correct class proto-

types. The comparison with KCL and KGen further shows that the gains are not obtained simply by adding a knowledge graph to a few-shot NER model. KCL uses KG information together with contrastive learning to improve representation discrimination, and KGen uses KG-derived class features from both the input-sentence side and the label-semantic side to strengthen semantic matching. These methods confirm the usefulness of external knowledge, but their main role is still representation enrichment. By contrast, KG-APC uses the KG prior after boundary-aware span filtering to correct unstable episode-level prototypes through a *K*-aware gate. This design is more directly aligned with the two dominant error sources in industrial few-shot NER: incomplete candidate spans and unreliable prototypes under extremely limited support evidence. Compared with LLM-assisted boundary-aware methods such as BANER [43], KG-APC remains lightweight and trainable, while still using external semantic knowledge through graph encoding. The Few-NERD results show a balanced picture: KG-APC is strongest in several one-shot settings, whereas BANER remains highly competitive in some five-shot settings. This suggests that the proposed correction mechanism is most valuable when support evidence is sparse, while larger support sets or additional LLM checking can narrow the gap on general-domain benchmarks.

The general-benchmark results should therefore be interpreted carefully. Across the 11 reported Few-NERD and CoNLL-2003 configurations, KG-APC obtains the numerically highest score in only three settings, all on Few-NERD; moreover, its 0.07-point advantage in Few-NERD INTRA 5-way 1-shot is not claimed to be statistically significant. BANER is numerically strongest in the other eight settings, including all three CoNLL-2003 configurations and all four Few-NERD 5-shot configurations. These results rule out a claim of universal superiority over LLM-assisted boundary checking. Two factors may explain the setting-dependent pattern. First, as the support size increases, textual prototypes become more reliable and the marginal value of KG correction decreases. Second, broad open-domain labels, especially the four CoNLL-2003 types, require less fine-grained type-semantic correction, whereas BANER can exploit broad LLM knowledge for candidate verification. KG-APC's main advantage is instead observed in the target industrial maintenance scenarios, where sparse supervision, specialized label semantics, and difficult maintenance spans align more closely with its design. This interpretation is also bounded by baseline provenance: the Few-NERD values follow the published BANER protocol and are not treated as locally paired-seed significance comparisons.

The dataset-specific trends further support the design choices of the framework. On MaintIE coarse-grained NER, the gains are stable because the five top-level classes still require reliable span selection and semantic separation. On MaintIE fine-grained NER, the improvements are larger, which is consistent with the fact that episodes are sampled from 32 semantically close intermediate types. In this setting, Wikidata aliases, descriptions, superclass relations, and related concepts provide useful type-level anchors for correcting noisy prototypes [3,4]. CFDK shows a different but complementary challenge. Although it contains only five categories, the difficulty varies greatly across labels: Equipment is usually easier to recognize, whereas Cause and Troubleshooting Solution often appear as long and heterogeneous sequences [5]. This explains why the boundary threshold and boundary-aware loss weight have different optimal values on CFDK and MaintIE fine-grained NER. The graph-hop analysis also shows that a two-hop neighborhood provides the best trade-off: one-hop aggregation may be too local, while three-hop aggregation can introduce weakly related neighbors and reduce prototype discriminability.

These findings have practical implications for industrial maintenance information systems. In real deployments, annotated maintenance data are often scarce, and entity schemas may be updated as equipment, fault modes, or maintenance procedures change. A framework that can combine a small support set with structured external knowledge is

therefore attractive for reducing annotation cost while preserving adaptability. However, the method also has clear limitations. Its effectiveness depends on the quality of label-to-Wikidata alignment and on whether the retrieved aliases, descriptions, and graph neighborhoods are semantically appropriate for the target schema. When an industrial entity type is highly specialized or poorly covered by Wikidata, the KG prior may become weak or noisy. Future work should therefore focus on two directions: incorporating hierarchical entity structures into prototype construction, and developing more adaptive mechanisms for assessing KG confidence and filtering noisy graph relations.

The perturbation results in Section 4.7 make this dependency more concrete. Under a 50% neighbor-replacement probability, KG-APC remains above the w/o KG variant by 0.26 F1 on MaintIE fine-grained NER and 0.45 F1 on CFDK. At a 50% type-alignment masking probability, it retains 58.05 and 54.42 F1 on the two datasets, respectively. These trends indicate graceful degradation under the evaluated conditions rather than complete dependence on a perfect KG. They also confirm that KG quality and coverage remain performance factors: both perturbation series decline monotonically as corruption increases. At the same 50% masking probability, KG-APC Full is numerically higher than w/o Gate by 0.69 F1 on MaintIE fine-grained NER and 0.64 F1 on CFDK. This controlled comparison links the adaptive gate to the observed performance at one representative incomplete-coverage point, while leaving its contribution across other perturbation levels unresolved.

The computational comparison adds a deployment-oriented dimension to the recognition results. As shown in Figures 2 and 3, KG-APC achieves 40.6–54.5-fold lower latency, 27.6–36.6-fold fewer FLOPs, and 6.8–7.8-fold lower peak memory than BANER across the three industrial settings. This advantage is mainly attributable to replacing the LLaMA-2-7B-scale inference backbone with RoBERTa-base and moving Wikidata neighborhood retrieval to an offline cache. Table 9 shows that even the 32-type MaintIE fine-grained schema requires only 22.7 s of one-time retrieval and a 17.9 MB cache. These measurements support the time- and space-efficiency claim under the reported RTX 6000 Ada FP16 environment, but they should not be interpreted as hardware-independent guarantees.

The remaining deployment limitation is narrower than simple random incompleteness. The new experiments perturb a static cached graph independently across neighbors or entity types; they do not test temporal graph drift, correlated errors across an equipment family, or repeated online cache updates. In practical maintenance environments, these structured changes may be more difficult than random missingness because several related type priors can become stale simultaneously. Future deployment studies should therefore evaluate concurrent workloads, dynamic cache refresh, confidence-calibrated KG fallback, and relation filtering under temporally evolving industrial schemas.

6. Conclusions

This paper investigates few-shot named entity recognition in industrial operation and maintenance scenarios, where annotated data are limited, entity boundaries are often ambiguous, and entity types are strongly connected to domain knowledge. To address these challenges, we propose KG-APC, a knowledge-graph-guided adaptive prototype correction framework. The framework first performs boundary-aware span detection to obtain more reliable candidate mentions, then uses external knowledge graph priors to enrich label semantics and correct class prototypes, and finally conducts metric-based classification with corrected prototypes. In this way, KG-APC jointly considers span boundary quality, support-set evidence, and structured domain knowledge, which are all important for few-shot industrial NER.

The experimental results demonstrate the effectiveness of the proposed framework. On the industrial maintenance datasets MaintIE and CFDK, KG-APC consistently achieves

the best performance under different N -way K -shot settings. The gains are especially meaningful for MaintIE fine-grained NER and CFDK, where entity types are semantically close or expressed as long diagnostic sequences. Compared with the knowledge-enhanced baselines KCL and KGen, KG-APC obtains more stable improvements on the industrial datasets, indicating that adaptive prototype correction and boundary-aware span modeling provide benefits beyond KG-based representation enrichment alone. The general-domain evidence is more limited: KG-APC is numerically best in three of the eleven Few-NERD and CoNLL-2003 configurations, while BANER leads in the remaining eight. We therefore conclude that KG-APC is competitive rather than uniformly superior outside its target industrial domain. The ablation studies verify that the KG prior, adaptive prototype correction gate, and boundary-aware span detection all contribute to the final performance. The hyperparameter analysis also shows that KG-APC is relatively stable under different support sizes, graph aggregation depths, boundary thresholds, and boundary-aware loss weights. The KG perturbation experiments further show gradual rather than abrupt degradation under random neighbor noise and missing type alignments, while delimiting temporal and systematically correlated KG changes as unresolved deployment conditions.

There are still two directions worth exploring in future work. First, many industrial entity schemas have explicit hierarchical structures, such as coarse categories, intermediate types, and fine-grained entity labels. Future models could directly incorporate such hierarchical entity structures into prototype construction and decision making, so that predictions at different granularity levels can mutually constrain and support each other. Second, the quality, coverage, and update frequency of external knowledge graphs may affect the reliability of KG-enhanced prototypes. A more adaptive mechanism for assessing KG confidence and filtering noisy relations would make the framework more robust in real industrial applications.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/electronics15153275/s1>: Table S1: Complete MaintIE 32-type mapping to Wikidata anchors and representative retained two-hop neighborhoods; Note S1: Detailed KG-prior construction and scale information and complete protocols for noisy-neighborhood replacement and missing entity-type alignment experiments.

Author Contributions: Conceptualization, P.D. and X.G.; methodology, P.D.; validation, P.D., X.G., and Y.X.; formal analysis, P.D.; investigation, P.D.; data curation, P.D.; writing—original draft preparation, P.D.; writing—review and editing, X.G. and Y.X.; visualization, P.D.; supervision, Y.X.; project administration, Y.X.; funding acquisition, Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 72071145.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All datasets analyzed in this study are publicly available. In particular, the original MaintIE and CFDK dataset papers cited in this manuscript provide their public repository addresses and access information. Few-NERD and CoNLL-2003 are likewise available from their cited public sources. No IMER dataset or private industrial subset was used in this study.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

KG-APC	Knowledge Graph-Guided Adaptive Prototype Correction
BA-SD	Boundary-Aware Span Detection
BERT	Bidirectional Encoder Representations from Transformers
GAT	Graph Attention Network
KG	Knowledge Graph
NER	Named Entity Recognition

References

- Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; de Melo, G.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge graphs. *ACM Comput. Surv.* **2021**, *54*, 1–37. [\[CrossRef\]](#)
- Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 494–514. [\[CrossRef\]](#) [\[PubMed\]](#)
- Vrandečić, D.; Krotzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM* **2014**, *57*, 78–85.
- Bikaun, T.K.; French, T.; Stewart, M.; Liu, W.; Hodkiewicz, M. MaintIE: A fine-grained annotation schema and benchmark for information extraction from maintenance short texts. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING), Torino, Italy, 20–25 May 2024*; ELRA: Torino, Italy; ICCL: Torino, Italy, 2024; pp. 10939–10951.
- Chen, T.; Zhu, J.; Zeng, Z.; Jia, X. Compressor fault diagnosis knowledge: A benchmark dataset for knowledge extraction from maintenance log sheets based on sequence labeling. *IEEE Access* **2021**, *9*, 59394–59405. [\[CrossRef\]](#)
- Xu, C.; Wang, Z.; Jin, Y.; Nong, W. An adaptive industrial large language model for mechanical fault diagnosis under variable operating conditions. *Adv. Eng. Inform.* **2026**, *74*, 104821. [\[CrossRef\]](#)
- Nadeau, D.; Sekine, S. A survey of named entity recognition and classification. *Lingu. Investig.* **2007**, *30*, 3–26. [\[CrossRef\]](#)
- Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 50–70. [\[CrossRef\]](#)
- Lafferty, J.; McCallum, A.; Pereira, F.C.N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML), Williamstown, MA, USA, 28 June–1 July 2001*; Morgan Kaufmann: San Francisco, CA, USA, 2001; pp. 282–289.
- Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
- Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), San Diego, CA, USA, 12–17 June 2016*; Association for Computational Linguistics: Kerrville, TX, USA, 2016; pp. 260–270.
- Ma, X.; Hovy, E. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), Berlin, Germany, 7–12 August 2016*; Association for Computational Linguistics: Kerrville, TX, USA, 2016; pp. 1064–1074.
- Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Minneapolis, MN, USA, 2–7 June 2019*; Association for Computational Linguistics: Kerrville, TX, USA, 2019; pp. 4171–4186.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
- Hu, Z.; Hou, W.; Liu, X. Deep learning for named entity recognition: A survey. *Neural Comput. Appl.* **2024**, *36*, 8995–9022. [\[CrossRef\]](#)
- Kamath, G.; Vajjala, S. Does synthetic data help named entity recognition for low-resource languages? In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL), Mumbai, India, 20–24 December 2025*; The Association for Computational Linguistics: Kerrville, TX, USA; The Asian Federation of Natural Language Processing: Singapore, 2025; pp. 159–167.
- Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), Barcelona, Spain, 5–10 December 2016*; Curran Associates, Inc.: Red Hook, NY, USA, 2016.

18. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
19. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.S.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 18–22 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1199–1208.
20. Fritzler, A.; Logacheva, V.; Kretov, M. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC)*, Limassol, Cyprus, 8–12 April 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 993–1000.
21. Hou, Y.; Che, W.; Lai, Y.; Zhou, Z.; Liu, Y.; Liu, H.; Liu, T. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Online, 5–10 July 2020; Association for Computational Linguistics: Kerrville, TX, USA, 2020; pp. 1381–1393.
22. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: Red Hook, NY, USA, 2017.
23. Yang, Y.; Katiyar, A. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, 16–20 November 2020; Association for Computational Linguistics: Kerrville, TX, USA, 2020; pp. 6365–6375.
24. Das, S.S.S.; Katiyar, A.; Passonneau, R.J.; Zhang, R. CONTAiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Kerrville, TX, USA, 2022; pp. 6338–6353.
25. Ma, T.; Jiang, H.; Wu, Q.; Zhao, T.; Lin, C.-Y. Decomposed meta-learning for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Kerrville, TX, USA, 2022.
26. Wang, J.; Han, C.; Wang, C.; Tan, C.; Qiu, M.; Huang, S.; Huang, J.; Gao, M. SpanProto: A two-stage span-based prototypical network for few-shot named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Abu Dhabi, United Arab Emirates, 7–11 December 2022; Association for Computational Linguistics: Kerrville, TX, USA, 2022.
27. Li, Y.; Yu, Y.; Qian, T. Type-aware decomposed framework for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore, 6–10 December 2023; Association for Computational Linguistics: Kerrville, TX, USA, 2023.
28. Zhang, S.; Cao, B.; Fan, J. KCL: Few-shot named entity recognition with knowledge graph and contrastive learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, Torino, Italy, 20–25 May 2024; ELRA: Torino, Italy; ICCL: Torino, Italy, 2024; pp. 9681–9692.
29. Nie, B.; Shao, Y.; Wang, Y. What can knowledge graph do for few-shot named entity recognition. *J. Web Semant.* **2025**, *86*, 100866. [[CrossRef](#)]
30. Ding, N.; Xu, G.; Chen, Y.; Wang, X.; Han, X.; Xie, P.; Zheng, H.-T.; Liu, Z. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, Online, 1–6 August 2021; Association for Computational Linguistics: Kerrville, TX, USA, 2021; pp. 3198–3213.
31. Sang, E.T.K.; De Meulder, F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL)*, Edmonton, AB, Canada, 31 May–1 June 2003; Association for Computational Linguistics: Kerrville, TX, USA, 2003; pp. 142–147.
32. Wiseman, S.; Stratos, K. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Kerrville, TX, USA, 2019; pp. 5363–5369.
33. Ma, J.; Ballesteros, M.; Doss, S.; Anubhai, R.; Mallya, S.; Al-Onaizan, Y.; Roth, D. Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Kerrville, TX, USA, 2022; pp. 1956–1971.
34. Huang, Y.; He, K.; Wang, Y.; Zhang, X.; Gong, T.; Mao, R.; Li, C. COPNER: Contrastive learning with prompt guiding for few-shot named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 2515–2527.
35. Ma, R.; Lin, Z.; Chen, X.; Zhou, X.; Wang, J.; Gui, T.; Zhang, Q.; Gao, X.; Chen, Y.W. Coarse-to-fine few-shot learning for named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, ON, Canada, 9–14 July 2023; Association for Computational Linguistics: Kerrville, TX, USA, 2023; pp. 4115–4129.

36. Wang, P.; Xu, R.; Liu, T.; Zhou, Q.; Cao, Y.; Chang, B.; Sui, Z. An enhanced span-based decomposition method for few-shot sequence labeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Seattle, WA, USA, 10–15 July 2022*; Association for Computational Linguistics: Kerrville, TX, USA, 2022; pp. 5012–5024.
37. Ji, S.; Kong, F. A novel three-stage framework for few-shot named entity recognition. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING), Torino, Italy, 20–25 May 2024*; ELRA: Torino, Italy; ICCL: Torino, Italy 2024; pp. 1293–1305.
38. Xue, X.; Zhang, C.; Xu, T.; Niu, Z. Robust few-shot named entity recognition with boundary discrimination and correlation purification. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI), Vancouver, BC, Canada, 20–27 February 2024*; AAAI Press: Washington, DC, USA, 2024; pp. 19341–19349.
39. Chen, P.; Wang, J.; Lin, H.; Zhao, D.; Yang, Z. Few-shot biomedical named entity recognition via knowledge-guided instance generation and prompt contrastive learning. *Bioinformatics* **2023**, *39*, btad496. [[CrossRef](#)] [[PubMed](#)]
40. Nie, B.; Shao, Y.; Wang, Y. Know-Adapter: Towards knowledge-aware parameter-efficient transfer learning for few-shot named entity recognition. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING), Torino, Italy, 20–25 May 2024*; ELRA: Torino, Italy; ICCL: Torino, Italy 2024; pp. 9777–9786.
41. Ok, H.; Kil, T.; Seo, S.; Lee, J. SCANNER: Knowledge-enhanced approach for robust multi-modal named entity recognition of unseen entities. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Mexico City, Mexico, 16–21 June 2024*; Association for Computational Linguistics: Kerrville, TX, USA, 2024; pp. 7725–7737.
42. Chen, W.; Zhao, L.; Zheng, Z.; Xu, T.; Wang, Y.; Chen, E. Double-Checker: Large language model as a checker for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, FL, USA, 12–16 November 2024*; Association for Computational Linguistics: Kerrville, TX, USA, 2024; pp. 3172–3181.
43. Guo, Q.; Dong, Y.; Tian, L.; Kang, Z.; Zhang, Y.; Wang, S. BANER: Boundary-aware LLMs for few-shot named entity recognition. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING), Abu Dhabi, United Arab Emirates, 19–24 January 2025*; Association for Computational Linguistics: Kerrville, TX, USA, 2025; pp. 10375–10389.
44. Ashok, D.; Lipton, Z.C. PromptNER: Prompting for named entity recognition. *arXiv* **2023**, arXiv:2305.15444.
45. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), Online, 6–12 December 2020*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 1877–1901.
46. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17–22 June 2006*; IEEE: Piscataway, NJ, USA, 2006; pp. 1735–1742.
47. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. In *Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018*.
48. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019*.
49. Paulheim, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semant. Web* **2017**, *8*, 489–508.
50. Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; Tang, J. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Online, 23–27 August 2020*; Association for Computing Machinery: Kerrville, TX, USA, 2020; pp. 66–74.

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.