Neural Networks 185 (2025) 107172

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

SQGE: Support-query prototype guidance and enhancement for few-shot relational triple extraction

Chen Gao [a], Xuan Zhang [b,c,*], Zhi Jin [d,e], Wei Cai [b], Danyang Wang [b], Kunpeng Du [a], Chunlin Yin [a], Tong Li [f,**]

[a] School of Information Science and Engineering, Yunnan University, Yunnan 650091, China
[b] School of Software, Yunnan University, Yunnan 650091, China
[c] Key Laboratory of Software Engineering of Yunnan Province, Yunnan 650091, China
[d] Key Lab. of High-Confidence Software Technologies (MoU), Peking University, Beijing 100871, China
[e] School of Computer Science, Peking University, Beijing 100871, China
[f] School of Big Data, Yunnan Agricultural University, Yunnan, 650201, China

ARTICLE INFO

Keywords:
Few-shot learning
Relational triple extraction
Prototype guidance
Feature enhancement
Contrastive learning

ABSTRACT

The current few-shot relational triple extraction (FS-RTE) techniques, which rely on prototype networks, have made significant progress. Nevertheless, the scarcity of data in the support set results in both intra-class and inter-class gaps in FS-RTE. Instances with restricted support sets make capturing the various features of target instances in the query set difficult, resulting in intra-class gaps. The support set lacks discernible target category characteristics, and the distances between data from various categories are insufficient, leading to intra-class gaps. In this paper, we propose an FS-RTE method based on support-query prototype guidance and enhancement (SQGE). It includes a support-query prototype guide module, which creates query prototypes based on the support prototype and combines the two prototypes. The fusion prototype can accurately capture the fundamental feature that aligns with the query set, suitably match the query features, and reduce the intra-class gap. Furthermore, to address the inter-class gap, we employ entity-level feature enhancement to improve the feature representation of target entities belonging to the same class. On the other hand, we construct positive and negative instances of the target class through contrastive learning, which not only strengthens the representation of the same target class but also distinguishes the feature space of the target class from other classes. Extensive experimental results on three datasets demonstrate the effectiveness of our approach. All the code and data are made available in https://github.com/gao929165733/SQGE_code.

1. Introduction

As a critical task in information extraction (Li, Sun, Han, & Li, 2020; Sarawagi et al., 2008; Wang, Cao, De Melo, & Liu, 2016), relation triple extraction plays a vital role in constructing knowledge graphs (Hogan et al., 2021). A relational triplet consists of entities and relations, which describe the relation between two entities. For example, give the sentence "*Jay Chou is a famous Chinese singer and actor.*", we can extract the triple ⟨"*Jay Chou*", "*nationality*", "*China*"⟩, where "*nationality*" is the relation between head entity "*Jay Chou*" and tail entity "*China*".

Significant results have been obtained with supervised learning-based RTE methods (Cunningham, Cord, & Delany, 2008), such as TPLinker (Wang, Yu et al., 2020), CasRel (Wei, Su, Wang, Tian, & Chang, 2020), PRGC (Zheng et al., 2021), and OneRel (Shang, Huang, & Mao, 2022). Nevertheless, these techniques heavily depend on extensively annotated datasets and struggle to achieve satisfactory results in few-shot settings. Furthermore, these techniques are limited to identifying the types of entities and relations that exist in the dataset, resulting in subpar performance when encountering unseen relations or entities, which makes it challenging to meet the needs of practical applications. Researchers proposed few-shot learning (FSL) (Wang, Yao, Kwok & Ni, 2020) to solve this problem. Few-shot relational triplet extraction (FS-RTE) (Cong et al., 2022; Fritzler, Logacheva, & Kretov, 2019; Wang et al., 2022; Yu, Zhang et al., 2020) aims to extract relational triples from a few labeled instances. Table 1 shows that the support set comprises two kinds of relations, each with two instances, and the query set contains one instance that needs to be categorized. This is a 2-way 2-shot FS-RTE task. The model is trained based on the

* Corresponding author at: School of Software, Yunnan University, Yunnan 650091, China.
** Corresponding author.
E-mail addresses: zhxuan@ynu.edu.cn (X. Zhang), tli@ynu.edu.cn (T. Li).

https://doi.org/10.1016/j.neunet.2025.107172
Received 29 August 2024; Received in revised form 21 December 2024; Accepted 13 January 2025
Available online 22 January 2025

Table 1
An illustration of 2-way 2-shot FS-RTE. The head entity is underlined and the tail entity is in wavy lines.

Support Set	
R1: Born_in	Instance1: Kobe Bean Bryant was born in Philadelphia. Instance2: Jay Chou, a famous singer from Taiwan, China.
R2: Capital_of	Instance1: The government of France operates primarily out of Paris, the nation's core. Instance2: Tokyo is the capital of Japan, serving as the country's political, economic, and cultural center.
Query Set	
R1 or R2	Jack is a pop singer from Canada who has released many famous songs.

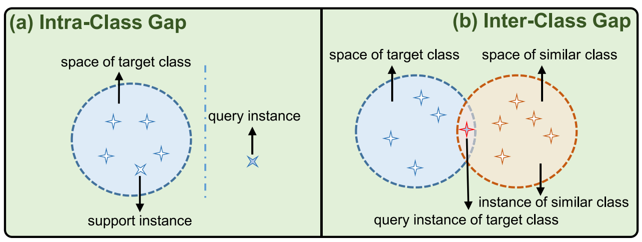

Fig. 1. Intra-class and inter-class gaps of FS-RTE. (a) The diversity of some samples in the same class leads to significant differences. (b) The target class may share features similar to those of a non-target class, causing inter-class gaps.

instances in the support set to identify the samples in the query set.

However, the number of instances for each class in the support set is limited under the few-shot setting. In contrast, there are numerous unknown instances in the query set that need to be recognized, and they contain many different types. The presence of a restricted number of support set instances poses a challenge in capturing the diverse features of the target instances in the query set, resulting in differences between the support features and query features of samples in the same class, leading to intra-class gaps. Furthermore, constructing prototypes from a limited number of instances in the support set makes creating an extensive representation of categories difficult. The features of the target class obtained in this way are not prominent enough, resulting in a considerable ambiguity between the target class and non-target classes with similar representations, and the distances between samples of different classes are not significant enough to lead to inter-class gaps. As shown in Fig. 1, such problems lead to bottlenecks in FS-RTE extraction performance.

Existing FS-RTE approaches improve performance by refining the quality of the prototype (Feng, Xu, Wang, Yang, & Huang, 2024; Fritzler et al., 2019; Gao, Han, Liu, & Sun, 2019; Han, Cheng, Wan, & Lu, 2023; Ji et al., 2022), utilizing external knowledge (Chen, Liu, Lin, Han, & Sun, 2022; Ma et al., 2022; Xiao, Jin, & Hao, 2021), and enhancing the interaction between support sets and query sets (Wang et al., 2022; Wen, Xia, Liao, & Tian, 2023). Gao et al. (2019) proposed a hybrid attention mechanism to enhance the prototype network and alleviate the impact of noisy samples on the model. Han et al. (2023) used relation label information to learn more informative and discriminative representations, thereby improving the performance of the model in handling difficult relations. Wen et al. (2023) used the hidden category information in the query set to generate more accurate relation prototypes to make up for the reliability of the support set prototypes. Although these methods have promoted the development of FS-RTE to a certain extent, their designs mainly focus on optimizing sample representation and similarity measurement. They are limited to one-sided attention to a certain type of technology in the FS-RTE task. It is better to analyze the root causes that affect the performance of the FS-RTE task from a holistic perspective, and pay in-depth attention to the intra-class and inter-class gaps that are unique to the FS-RTE task. The specific comparative analysis of these models is shown in Table 2. The core of this problem is that due to the limited support set data in the few-shot learning scenario, the diverse patterns that may appear in the

Table 2
Technical comparison and differences between SQGE and other existing models. "OP" refers to Optimizing prototypes, "UEK" refers to Utilizing external knowledge, "ETI" refers to Enhancing the interaction, and "ICICG" refers to solving Intra-class and inter-class gaps.

Model	OP	UEK	ETI	ICICG
Feng et al. (2024)	✓			
Fritzler et al. (2019)	✓			
Gao et al. (2019)	✓			
Han et al. (2023)	✓	✓		
Ji et al. (2022)	✓			
Xiao et al. (2021)		✓		
Ma et al. (2022)		✓		
Chen et al. (2022)		✓		
Wang et al. (2022)			✓	
Wen et al. (2023)			✓	
Our (SQGE)	✓	✓	✓	✓

query set cannot be effectively captured. In addition, feature differences between categories often lead to class confusion, affecting the accuracy of the model.

This study is the first to introduce the problem of intra-class and inter-class gaps into the research perspective of the FS-RTE task. We systematically analyze the impact of intra-class and inter-class gaps on model performance in few-shot learning and propose targeted solutions. We propose support-query prototype guidance and enhancement few-shot relation triple extraction method (SQGE). As shown in Fig. 2(a), traditional methods usually only use support sets to construct prototypes and use them to match relation instances in query sets. However, due to the scarcity of support set data, prototypes built solely based on support sets make it difficult to fully capture the complex and diverse feature patterns in query sets, which may lead to intra-class gaps when matching query sets and make it difficult to obtain ideal extraction results. As shown in Fig. 2(b), SQGE innovatively introduces a support-query prototype guidance module. Based on traditional prototype construction, it not only relies on the support set but also makes full use of the query set's information to generate a support-query fusion prototype. Through the joint action of this support and query set, we can capture the consistent basic features of the query object, making the constructed prototype more representative, thereby improving the adaptability and matching accuracy of the prototype to the query set features. We hypothesize that incorporating some features of the instance itself from the query set can improve its identification, thus improving the performance of the model by combining the relevant features from both the support set and the query set.

Table 3 displays the impact of utilizing the query set prototype for identifying relational triples in the 5-way 1-shot scenario. It demonstrates that incorporating varying quantities of samples from the correct query set to construct the prototype yields significantly superior results compared to solely utilizing the support set. However, we lack access to the gold entity labels of the query set during the training process. Thus, we employ specific gold entity labels and introduce noise to replicate the query set labels acquired throughout the training phase, which still results in improved performance compared to the baseline. It can be seen that using query set instances to construct prototypes can better represent the features of relevant entities in the query set, which is

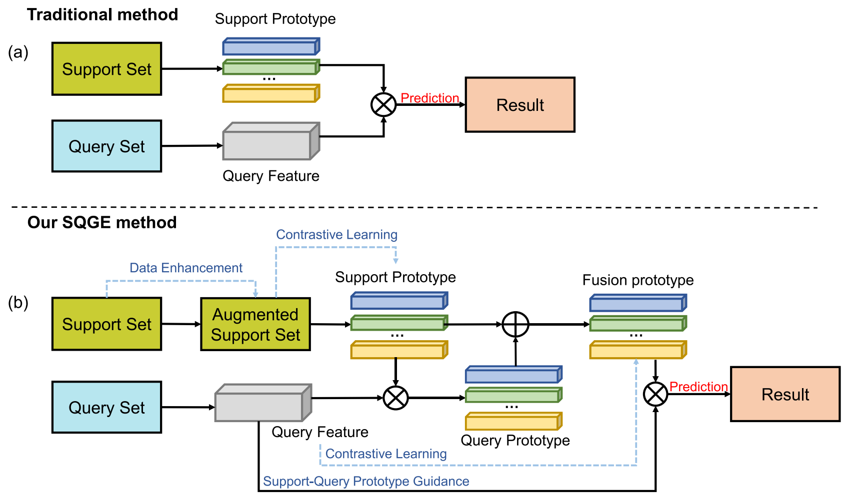

Fig. 2. Comparison of our proposed framework with previous frameworks.

Table 3
Effect of the query set prototype under 5-way 1-shot on FewRel dataset. SP denotes the support prototype, QP denotes the query prototype, and False denotes the addition of error samples. For example, 1False denotes the addition of an error sample.

Setting	SP	SP+50%QP	SP+33%QP	1False+SP+QP	2False+SP+QP	3False+SP+QP
F1 Score	32.85	53.47	40.94	55.93	52.47	48.37

directly related to the quality of the instances used.

Due to the limited support set data, it is extremely difficult to obtain sufficient and representative features of the target class. This scarcity makes it difficult to effectively form the feature space of the target class, and the features of the non-target class may be highly overlapped, resulting in insufficient inter-class separability. This phenomenon makes it difficult for the model to accurately distinguish the target class from other categories when matching the query set. Therefore, to address this problem, we propose a strategy that combines entity-level feature enhancement with multi-level contrastive learning. Specifically, we expand the features of entities in the support set through entity-level feature enhancement to more comprehensively characterize their entity types. This method reduces the ambiguity between categories by increasing the refinement and enrichment of the target class entity features to form a more representative feature distribution in the support set. In addition, we further introduce multi-level contrastive learning to construct positive and negative sample pairs of the target class from the dual perspectives of the support set and the query set. In this process, we not only construct contrast pairs in the support set but also perform feature contrast in the query set to enhance the distinguishability of the target class in the feature space. Through this multi-level contrastive learning mechanism, the feature representation of the target class is effectively enhanced while maintaining a clear separation from other classes in the feature space so that the model can show higher accuracy and robustness when dealing with inter-class gaps in few-shot scenarios.

We summarize the contributions of the paper as follows:

1. To alleviate the intra-class gap, we build the support-query fusion prototype to capture the consistent basic features of the query object and appropriately match the query characteristics.
2. To alleviate the inter-class gap, we propose a method that combines entity-level feature enhancement and multi-level contrastive learning to enhance the feature representation of the target class and separate the feature space of the target class from other classes.
3. We perform comprehensive experiments on public datasets and compare the results with state-of-the-art models. Experimental results show that SQGE significantly improves the F1 score by 0.9%∼16.2% on FewRel, FewNYT and TACRED.

2. Related work

2.1. Relational triple extraction

Relational triple extraction (Sarawagi et al., 2008) is a fundamental task in the construction process of knowledge graphs (Hogan et al., 2021), which extracts the entities and relations present in a sentence. Early research focused on pipeline-based methods, divided into named entity recognition (Chiu & Nichols, 2016; Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016; Mansouri, Affendey, & Mamat, 2008) and relation extraction (Zeng, Liu, Lai, Zhou, & Zhao, 2014; Zhou et al., 2016). The technique is relatively simple but has problems such as error propagation (Tan, Zhao, Wang, & Xiao, 2019; Wang, Yu et al., 2020). To alleviate the issues existing in pipeline-based methods, some works proposed joint extraction models (Zhao, Yan, Cao, & Li, 2021). RTE methods based on supervised learning (Ning, Yang, Sun, Wang, & Lin, 2023; Shang et al., 2022; Wang, Yu et al., 2020; Wei et al., 2020; Zheng et al., 2021) have shown satisfactory performance but still have significant limitations. Supervised learning is typically labor-intensive in a specific domain due to the need for a substantial amount of labeled data. Therefore, researchers have started to investigate FS-RTE.

2.2. Few-shot learning

Few-Shot Learning (FSL) (Wang, Yao, Kwok & Ni, 2020) aims to train a model with only a few labeled training samples, which can predict new tasks well. Early research in FSL focused on computer vision, such as image classification (Tian, Wang, Krishnan, Tenenbaum, & Isola, 2020), and subsequently expanded to encompass the field of natural language processing (Sun, Sun, Zhou, & Lv, 2019). There are three main categories in existing FSL approaches. The first is based on transfer learning (Zhuang et al., 2020), where knowledge learned from one task is transferred to another related or different task to improve performance on the target task. The second approach is based on meta-learning (Hospedales, Antoniou, Micaelli, & Storkey, 2021), which rapidly updates the model through meta-learning optimization using

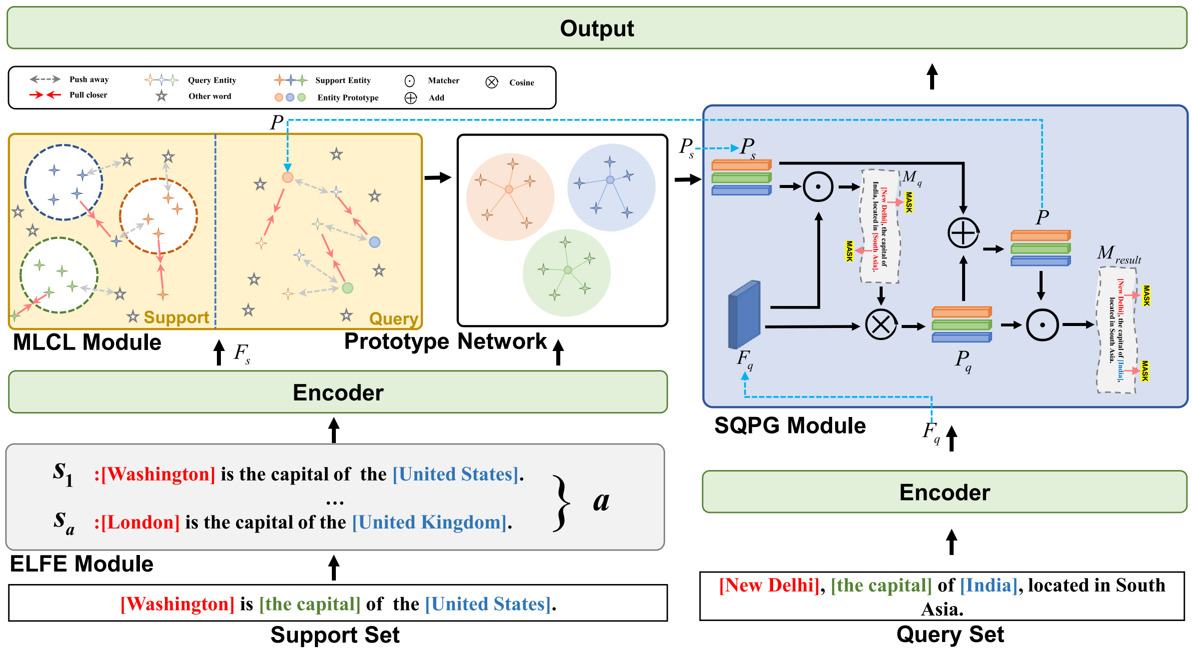

Fig. 3. Overview of FS-RTE framework with SQGE.

limited samples. The third approach relies on metric learning (Snell, Swersky, & Zemel, 2017), which employs the similarity measure to calculate the similarity values between the query and support features.

2.3. Few-shot relational triple extraction

Few-shot relational triple extraction identifies relational triples by learning from small amounts of labeled data. Similar to the RTE method based on supervised learning, it can also be performed in two stages, which are few-shot named entity identification (FS-NER) (Das, Katiyar, Passonneau, & Zhang, 2022; Feng et al., 2024; Huang et al., 2021; Ji et al., 2022; Ma et al., 2022; Tian et al., 2020) and few-shot relation extraction (FS-RE) (Dong et al., 2020; Gao et al., 2019; Han et al., 2023; Wen et al., 2023; Xiao et al., 2021; Ye & Ling, 2019). These approaches provide ideas for FS-RTE solutions. However, they only consider one task and treat each as an individual model. This approach does not address the underlying challenges faced by FS-RTE. As a result, researchers suggested utilizing a joint extraction strategy for conducting FS-RTE (Cong et al., 2022; Wang et al., 2022; Yu, Zhang et al., 2020).

Yu, Zhang et al. (2020) first proposed using a model to simultaneously extract entities and relations in sentences, but there are some limitations. Using CRF directly for NER in the FS-RTE is challenging since there is insufficient labeled data to reach the desired performance. Furthermore, the extracted incorrect entities will influence the results of relation extraction. Cong et al. (2022) employed a relation-guided methodology to perform triple extraction. They created relation prototypes to initially detect relations in sentences and subsequently developed entity prototypes based on these relations to identify entities in sentences. Wang et al. (2022) introduce a translation-based graph reasoning network that combines the translation model with graph reasoning, significantly advantages handling tasks with complex dependencies. In their study, He, Song, Cheng, and Xu (2022) used the nearest neighbor matching (Dang, Deng, Yang, Wei, & Huang, 2021) method to evaluate the semantic similarity of words to obtain relation triplets in sentences. Fei, Zeng, Zhao, Li, and Xiao (2022) introduced a new perspective transfer network that effectively utilizes global information for addressing FS-RTE. By thoroughly mining the local and global information in the phrase, it extracts relation triples based on three viewpoints: relation, entity, and triple perspectives. Jiang, Zhu, and He (2023) propose a token-level FS-RTE (TLSM), which is a method based on semantic similarity information. In order to solve the problem of error propagation, RCTE (Liao, Lu, & Guo, 2024) proposes a relation candidate-guided FS-RTE.

The joint extraction paradigm is significantly more challenging than the pipeline method due to the simultaneous consideration of entity and relation representations. This becomes particularly problematic in few-shot scenarios, necessitating the exploration of more effective extraction strategies. Furthermore, while these approaches have had some success in optimizing matching methods and prototype refinement efforts, none of them considers intra-class and inter-class gaps in FS-RTE.

3. Problem definition

Give a sentence $S = \{s_1, s_2, \dots, s_i\}_{i=1}^{m}$ and relation set $R = \{r_1, r_2, \dots, r_j\}_{j=1}^{k}$, RTE model is aim to detect relational triples $T = \{\langle h_i, r_j, t_i \rangle \mid h_i, t_i \in E, r_j \in R\}$, where E denotes the set of entities, h_i and t_i denote the head and tail entities, respectively. We split all instances into two datasets, D_{train} and D_{test}. It is important to mention that the relation sets R_{train} and R_{test} do not overlap.

Following the previous meta-learning paradigm, we adopt the "episode" training strategy (Yu, Ji, Han & Zhang, 2020). D_{train} and D_{test} are divided into $\{D_{train}^{support}, D_{train}^{query}, D_{test}^{support}, D_{test}^{query}\}$. In the training episode, we construct the train support set $D_{train}^{support} = \{(s_i, \langle h_i, r_i, t_i \rangle | r_i \in R_{train}, h_i, t_i \in E_{train})\}_{i=1}^{NK}$ and train query set $D_{train}^{query} = \{(s_i, \langle h_i, r_i, t_i \rangle \mid r_i \in R_{query}, h_i, t_i \in E_{query})\}_{i=1}^{NG}$ by randomly selecting N classes with K and G instances for each class. $D_{test}^{support}$ and D_{test}^{query} are constructed in the same way. We refer to such an FSL problem as the N-way K-shot problem.

4. Methodology

The model consists of three core modules, which are the entity-level feature enhancement module, the multi-level comparative learning module and the support-query prototype guidance module. The entity-level feature enhancement and multi-level comparison learning modules are primarily utilized to resolve the inter-class gap. They enhance the depiction of the target entity category and distinguish it from the class that shares similar characteristics. The support for

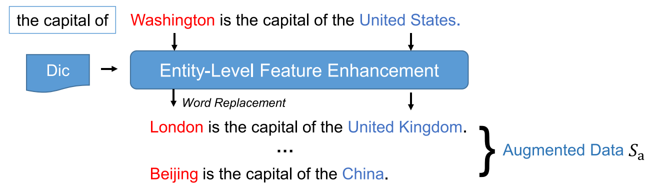

Fig. 4. Entity-level feature enhancement (ELFE) in sentences.

the query prototype boot module focuses on the intra-class gap. It matches the query properties appropriately by capturing the essential characteristics of the support-query object consistency, reducing the differences between the attributes within a class.

Fig. 3 show the comprehensive structure of SQGE. The input to the model consists of the support set and query set data, which are processed by an encoder to extract features. Among them, in the entity-level feature enhancement module, we use support to concentrate on the same type of entity instances to enhance the representation of the current entity type. The specific method randomly selects entities of the same kind from the support set to replace the entities in the sentence, forming a new sequence. In addition, positive and negative instances of the target entity class are constructed through contrastive learning modules. Entities of the same type are regarded as positive example pairs, other entities and non-entities are considered negative example pairs. This method can separate the feature space of the target class from other classes while enhancing the representation of the same target entity type. The entity-enhanced support set data is utilized to construct a support prototype. This support prototype is then employed to forecast the query set, resulting in the acquisition of the query set initial mask. Subsequently, the query prototype is constructed using the query set and query mask. We combine the two types of prototypes to form the final prototype and identify the triples existing in the sentence. By incorporating relations into the entity labels, our task is to construct representations of the entity prototypes that directly specify the head and tail entities associated with the respective relations, resulting in triples.

4.1. Entity-level feature enhancement module

Prototype network-based methods only use small data for prototype construction under few-shot. However, the characteristics of a few entities make it difficult to represent the entity type. The prototype constructed in this way has large deviations and is inaccurate. A large amount of data based on the open domain is proposed to be obtained in certain studies, but getting more label data is expensive and goes against the initial goals of few-shot learning. While in the support set, there are some other instances for a type of data, and we can use these instances for feature enhancement. Based on this, we propose an entity-level feature enhancement method to improve entity representation in sentences. The support set has multiple sentence instances under each type of relation. Each sentence contains both head and tail entities. Taking a 5-way 1-shot as an example, the input of each episode includes five sentences, each of which belongs to a different relation category. Given the support set and a query set as input, we expand the support set of the original input using an entity-level feature enhancement-based approach. As shown in Fig. 4.

We construct the relation-entity dictionary based on the support set. For example, under the relation "*the capital of*", the sentence "*Washington is the capital of the United States.*" contains the head entity "*Washington*" and the tail entity "*United States*". We keep the backbone of the sentence unchanged and randomly replace the head and tail entities in the sentence. The replaced entities are other entities of the same length under the same relation to form a new sequence. To prevent data leakage, the entities replaced under each episode do not include entities that appear in the query set. When the entity feature enhancement is completed, we use the BERT (Kenton & Toutanova, 2019) encoder to extract features from the data to obtain the support set features $F_s = \{w_1^s, w_2^s, \dots, w_n^s\}$ and query set features $F_q = \{w_1^q, w_2^q, \dots, w_n^q\}$ respectively. The formula is as follows:

$$S_a = EntityAug(S) \tag{1}$$

$$F_s, F_q = BERT(\{s_1, s_2, \dots, s_a\}, \{q_1, q_2, \dots, q_n\}) \tag{2}$$

where $F_s \in {}^{N \times k_{aug} \times d_h}$ and $F_q \in {}^{N \times 1 \times d_h}$. S_a is the enhanced support set. N is the relation category. k_{aug} represents the number of sentences in each relation category after entity-level feature enhancement. d_h is the feature dimension, $EntityAug$ represents the method of entity feature enhancement.

4.2. Support-query prototype guidance module

Prototype network-based approaches primarily utilize the data in the support set to generate prototypes. The initial prototypes are obtained by averaging relevant instances in the support set and then matched with query features to make predictions. This approach is both intuitive and successful up to a certain degree. The prototype created by averaging many instances captures the shared characteristics among the samples. However, only a few instances are often selected as support sets during model training. The limited samples in the support set are insufficient to represent the semantic class of the target object in the complete query set, leading to the intra-class gap. Therefore, we propose a support-query prototype guidance module to fuse the prototypes and mitigate the problem of the intra-class gap between support and query entities in the same class to accommodate target recognition better.

Given the support set feature F_s and the query set feature F_q, we obtain the entity vector $E_s = \{e_1^s, e_2^s, \dots, e_{k_{aug}}^s\}$ of the support set based on the support entity mask M_s. Then calculate the support entity prototypes of each episode and obtain the support entity prototype set P_s, which is formulated by:

$$P_s^{c_j} = \frac{1}{k_{aug}} \sum_{i=1}^{k_{aug}} e_i^s, j = 1, 2, \dots, 2N+1 \tag{3}$$

$$P_s = \{P_s^{c_1}, P_s^{c_2}, \dots, P_s^{c_{2N+1}}\} \tag{4}$$

where k_{aug} is the number of sentences in each relation category after entity-level feature enhancement, c_i represents the category of the ith prototype. Since our model requires the construction of head and tail entity prototypes separately, there are $2N$ entity prototypes and one other class prototype for N class relations.

After obtaining the support set entity prototype, we can construct the query prototype P_q in the same way. However, during the training process, the query set's entity labels are unavailable. Therefore, we use a predicted query entity mask M_q to aggregate query features. The construction technique involves utilizing the support set entity prototype P_s to match with the query set sentence F_q to get the query entity mask M_q. Then the query set sentences F_q and query entity masks M_q can be aggregated to obtain the query entity prototype P_q. The formula is as follows:

$$M_q = \mathrm{Cos}ine(\mathrm{P}_s, \mathrm{F}_q) \tag{5}$$

$$\widetilde{M}_q = \mathrm{I}(\mathrm{M}_q > \tau) \tag{6}$$

$$\mathrm{P}_q = Aggregator(\mathrm{F}_q, \widetilde{\mathrm{M}}_q) \tag{7}$$

where Cosine is similarity function, I is indicator function. The mask threshold τ is used to control the range of query feature sampling. We choose the optimal experimental results to set it, Aggregator is aggregator function.

The query set entity prototype contains the characteristics of the query set. Still, the query entity mask is predicted based on the support set entity prototype, which is not entirely accurate. Therefore, to form our final entity prototype representation, we use a weighted approach to fuse the support set's entity prototypes with the query set's entity prototypes. To get the final matching prediction, we use a two-layer Convolutional Neural Network (CNN) (LeCun et al., 1989) along with a subsequent linear layer to compute the similarity score between the prototype and query characteristics. The formula is as follows:

$$\mathrm{P} = \alpha_1 P_s + \alpha_2 P_q \tag{8}$$

$$M_{result} = Lin(Conv([P; F_q; P - F_q; P + F_q; P \otimes F_q])) \tag{9}$$

where α_1 and α_2 is weighting factor, Lin and $Conv$ means the linear function and CNN, $\otimes$ means element-wise product, we set $\alpha_1 = \alpha_2 = 1$.

4.3. Multi-level contrastive learning module

We introduce a support and query-oriented multi-level comparative learning approach to mitigate inter-class gaps further. It builds positive and negative instances to compare and learn more about the different representations between different entity classes. Our contrastive learning loss is constructed using a supervised contrastive learning approach, which utilizes the label information of the data to build positive and negative pairs. The support-level contrastive learning (SLCL) specific construction process is as follows: at the support set level, for entity w_k in the support set under the relation i, we take the entities of the same class in s_i as positive pairs $P_{s(i)}$, and the other entities of different classes as well as the non-entity instances as negative examples $N_{s(i)}$. Contrastive learning seeks to minimize the distance between examples of the same class while maximizing the distance between instances of different classes. In this way, we increase the discriminative power of the support set instances and generate more compact clusters for better prototypes. The SLCL loss is computed as follows:

$$\mathcal{L}_{SLCL} = \sum_{i=1}^{NK} \mathcal{L}_{SLCL_i} \tag{10}$$

$$\mathcal{L}_{SLCL_i} = -\log \frac{\exp(w_i^s \cdot w_j^s / \mathcal{T})}{\sum_{m=1}^{NK} \mathrm{I}_{m \neq i} \exp(w_i^s \cdot w_m^s / \mathcal{T})} \tag{11}$$

where w_i^s and w_j^s represent the word feature vector representation in the support set, j means the same type of instance as i, and m means not the same type of instance as i. $\mathcal{T}$ is a temperature factor, N and K represent the specific settings in N-way K-shot.

At the query set level, for the set of entity prototypes P under the current episode, we use each prototype as an anchor point and consider query set entity instances e_k^q of the same type as positive pairs $P_{q(i)}$ and query instances of different types as negative pairs $N_{q(i)}$. The purpose of query-level contrastive learning (QLCL) is to optimize the prototypes based on the entity labels of the query set. The QLCL loss calculation formula is as follows:

$$\mathcal{L}_{QLCL} = \sum_{c \in P} \sum_{i=1}^{NG} \mathcal{L}_{QLCL_c^i} \tag{12}$$

$$\mathcal{L}_{QLCL_c^i} = -\log \frac{\exp(p_c \cdot e_i^q / \mathcal{T})}{\sum_{m=1, m \neq i}^{2NG} \exp(p_c \cdot e_m^q / \mathcal{T})} \tag{13}$$

$$\mathcal{L}_{CL} = \mathcal{L}_{SLCL} + \mathcal{L}_{QLCL} \tag{14}$$

where e_i^q and e_m^q represent the word feature vector representation in the query set, m means not the same type of instance as i. $\mathcal{T}$ is a temperature factor, N represent the category of relation and G is the number of query set instances under relation.

The loss of our model comprises two components, specifically contrastive learning loss and matching loss. The loss function of the training process is as follows:

$$\mathcal{L}_{Match} = \mathcal{L}_{CE}(M_{Result}, M_{True}) \tag{15}$$

$$\mathcal{L} = \mathcal{L}_{CL} + \mathcal{L}_{Match} \tag{16}$$

where $\mathcal{L}_{CE}$ is the binary cross-entropy loss function, M_{Result} is the predict result and M_{True} is the gold entity label.

4.4. Model training process

We describe the algorithmic flow in Algorithm 1. For the input of the support set, we first perform data enhancement on the support set according to the entity-level feature enhancement strategy and then encode them. Based on the data-augmented support set, we build the support entity prototype and obtain the predicted query entity mask to generate the query prototype similarly. Finally, the support and query prototypes are integrated to form the final prototype. We calculate the distance between the prototype and the query features to get the matching prediction. In this process, we compute the support-level contrastive learning loss using the original support set and the query-level contrastive loss using the prototype and query set entity instances.

Algorithm 1 The training process of SQGE.

Input: $D_{train}^{support} = \{(s_i, < h_i, r_i, t_i > | r_i \in R_{train}, h_i, t_i \in E_{train})\}_{i=1}^{NK}$; $D_{train}^{query} = \{(s_i, < h_i, r_i, t_i > | r_i \in R_{query}, h_i, t_i \in E_{query})\}_{i=1}^{NG}$
Output: $T = \{< h_i, r_j, t_i > | h_i, t_i \in E, r_j \in R\}$

```
Obtain enhanced support set S_a from D_train^support by Equation (1) in ELFE module;
Encode the input by Equation (2), obtain supports set features F_s and query set features F_q;
for episode in episodes do
    for i = 1 → N do
        Build support prototypes P_s by Equation (3) and (4);
        Calculate query masks M_q by Equation (5);
        Build query prototypes P_q by Equation (6) and (7);
        Fusion of support and query prototype to obtain final prototype P by Equation (8);
    end for
    Calculate query prediction loss L_match by Equation (15);
    Construct positive and negative sample pairs;
    Compute multi-level contrastive loss L_CL by Equation (14);
    Calculate L for this episode by Equation (16);
end for
Let L to be minimized in the next episode.
```

5. Experiment

5.1. Datasets

We mainly use the following datasets to evaluate the model. Table 4 displays comprehensive statistics of the datasets. The FewRel (Han et al., 2018) is highly prevalent in FS-RTE, comprising 100 relations and 70,000 sentences. To ensure a fair comparison with previous works, we follow the setting of Cong et al. (2022). Note that the relations classes in these datasets are independent. Based on the work (Fei et al., 2022), we constructed the FewNYT dataset. NYT dataset (Riedel, Yao, & McCallum, 2010) is a commonly used dataset for evaluating supervised RTE. It contains 24 relations, and there may be multiple relations in each sentence. We preprocessed the dataset according to the number of relations and instances to form a dataset containing 16 relations, which is used as the testing set. The training set and dev set are the same as FewRel. TACRED (Zhang, Zhong, Chen, Angeli, & Manning, 2017) is a large-scale human-annotated relation extraction dataset. Its data mainly comes from newswires and web texts used in the NIST TAC KBP slot-filling evaluation. We organized the dataset and constructed an FS-RTE dataset, in which the training set contains 15 types of relations, and the test set and dev set each contain 11 types of relations.

Table 4
Information statistics in FewRel, FewNYT and TACRED datasets.

Category	FewRel			FewNYT			TACRED		
	Dev	Test	Train	Dev	Test	Train	Dev	Test	Train
Relation	15	15	50	15	16	50	11	11	15
Entity	21,000	21,000	70,000	21,000	1,600	70,000	742	1238	24846
Sentence	10,500	10,500	35,000	10,500	800	35,000	371	619	12423

5.2. Experimental settings

The experimental conditions we used closely mirror those outlined in work (Gao et al., 2024). During the training method, we use the Adam (Kingma, 2014) optimizer with an initial learning rate of 1e-5 and a weight decay of 1e-3. We use the "episode" training methodology to train the model, whereby each training "episode" comprises $N \times K$ support instances and one query instance. The number of training iterations is set to 30,000, the number of test iterations is set to 3000, and the number of validation iterations is set to 500. We determine several threshold parameters in the model based on the experimental results reported in Section 5.4. The model evaluation approach follows previous research by using F1-score, precision, and recall to evaluate the model's performance.

5.3. Experimental results

We compared our model with the following baseline models. (1) CasRel (Wei et al., 2020): An approach for extracting overlapping triplets using supervised learning. (2) MatchNet (Vinyals, Blundell, Lillicrap, Wierstra, et al., 2016): It is founded on the concept of metric learning and employs external memory to augment the network, enhancing its learning capacity. (3) Proto Snell et al. (2017): A approach for few-shot learning utilizing prototype networks. (4) FS-GNN (Garcia & Bruna, 2018): It transforms few-shot learning into a supervised message passing task and trains it using a graph neural network. (5) MPE (Yu, Zhang et al., 2020): It introduces a unique multi-prototype embedding network to address the issue of few-shot relation triple extraction. By combining text representation learning and knowledge graph constraints, the model shows excellent generalization ability with a small amount of labeled data. (6) MLMAN (Ye & Ling, 2019): An approach enhances the conventional few-shot relation classification prototype network by iteratively encoding query instances and class prototypes using multi-level matching and aggregation. (7) NNM (He et al., 2022): An approach for extracting connection triplets from the text that combines few-shot learning methods with nearest-neighbor matching. (8) TGIN (Wang et al., 2022): A translation-based graph reasoning network combines the translation model with graph reasoning, significantly advantages handling tasks with complex dependencies. (9) PTN (Fei et al., 2022): It proposes a novel perspective transfer network, which verifies the extracted elements at local and global levels and effectively handles more realistic and challenging few-shot RTE scenarios by converting between perspectives of relations, entities, and triples. (10) RelATE (Cong et al., 2022): It presents a "Relation-then-Entity" task decomposition technique that first identifies the relations in the phrase using a double-layer attention mechanism, and then utilizes annotated samples of the identified relations to extract the matching entities. (11) TLSM (Jiang et al., 2023), an FS-RTE method based on token-level similarity of the entity label. (12) RCTE (Liao et al., 2024), a novel relation candidate-guided few-shot relational Triple Extraction approach.

Table 5 compares the F1-score results of SQGE with all baseline models on the FewRel dataset. Our proposed method SQGE achieved the best results. The experimental results of CasRel demonstrate that the model constructed using the conventional supervised learning paradigm cannot address the RTE task under the few-shot. MPE performs relation triplet extraction in stages. It uses a conventional entity extractor to identify entities in sentences. Under a small number of training instances, it is difficult to obtain an ideal entity recognition effect, which further affects the subsequent triplet extraction effect.

PTN, RelATE, RCTE and TLSM are the latest and most effective baseline models. We will focus on comparing these methods. On 5-way 5-shot, SQGE improves TLSM and RCTE by 1.3% and 2.1% respectively. On 10-way 10-shot, SQGE improves TLSM and RCTE by 0.9% and 3.3% respectively. However, TLSM and RCTE have only been experimented with in these two settings, so a more detailed comparison cannot be made. On 5-way 5-shot, 10-way 5-shot, and 10-way 10-shot, SQGE improved by 9.5%, 6.8%, and 10.2%, respectively, compared with PTN. On 5-way 1-shot and 10-way 1-shot, SQGE improved by 8.5% and 5.9%, respectively, compared with PTN. Compared to RelATE, SQGE showed improvements of 7.2%, 5.6%, and 5.5% on 5-way 5-shot, 10-way 5-shot, and 10-way 10-shot tasks, respectively. On 5-way 1-shot and 10-way 1-shot, SQGE is 9.8% and 10.0% higher than RelATE, respectively. RCTE proposes a relation candidate-guided few shot relation triple extraction method. It uses a gate mechanism and a beam search framework to optimize the error propagation problem in FS-RTE. However, it is still based on the relation-then-entity paradigm and cannot fundamentally solve this problem. SQGE unifies entity and relation extraction and directly constructs relation-entity prototypes, which can effectively avoid the error propagation problem in the relation-then-entity paradigm. TLSM proposes an entity tag Token-level similarity evaluation method, which uses the semantic similarity information between tokens in a small number of annotated samples and unseen samples to improve the accuracy of entity extraction, thereby improving the overall performance of relational triple extraction. However, the above two methods only predict triples in the query set based on the information of the support set, without considering the differences between the support set and the query set, and thus lack unified modeling of intra-class and inter-class features. PTN uses a perspective conversion network to create relation and entity prototypes, convert between relation, entity, and triple views, and fully use sentence local and global information. RelATE identifies relations present in sentences based on relational prototypes. They are all based on two-stage extraction of relational triples, which results in error propagation and affects the performance of the final extraction of relational triples. Furthermore, none of these methods consider the intra-class and inter-class gaps in the FS-RTE, whereas our proposed method can effectively solve these problems.

Table 6 gives the experimental results of the FewNYT dataset. Since only a few models use this dataset, we only list the available baselines. SQGE still achieves the best performance, especially in the 5-way 1-shot and 10-way 1-shot scenarios. Table 7 shows the experimental results of the TACRED dataset. To compare with the previous baseline, we added two additional scenarios: 3-way 1-shot and 3-way 5-shot. Since most of the data is manually annotated, TACRED is of higher quality than the other two datasets, and SQGE still maintains a huge performance advantage.

To analyze the performance of the model in each subtask, more specifically, four scenarios were selected to test the entity, relation, and triplet extraction performance of the model. From Table 8, we can see that the performance of relation extraction is significantly better than that of named entity recognition. Named entity recognition is more complicated than relation extraction in few-shot. SQGE has no advantage in single entity and relation extraction compared to other baselines, especially in entity recognition. We employ a joint approach to modeling the FS-RTE, utilizing a single-stage extraction architecture.

Table 5
The F1 scores of the compared model on the FewRel dataset. **Bold** indicates the best results, and underlined indicates the second-best results.

Model	5W1S	5W5S	10W1S	10W5S	10W10S
CasRel (Wei et al., 2020)	–	2.1	–	–	2.0
FS-GNN (Garcia & Bruna, 2018)	17.8	24.5	11.4	–	16.1
MatchNet (Vinyals et al., 2016)	15.4	18.7	8.2	–	16.3
MPE (Yu, Zhang et al., 2020)	–	23.3	–	–	12.1
Proto (Snell et al., 2017)	15.9	21.2	10.4	–	15.4
MLMAN (Ye & Ling, 2019)	20.4	28.5	15.3	–	19.2
TGIN (Wang et al., 2022)	24.0	32.3	17.3	–	22.8
NNM (He et al., 2022)	–	32.1	–	–	25.0
RelATE (Cong et al., 2022)	28.7	42.3	20.3	34.8	40.9
PTN (Fei et al., 2022)	30.0	40.0	25.3	33.6	36.2
RCTE (Liao et al., 2024)	–	45.3	–	–	41.2
TLSM (Jiang et al., 2023)	–	48.2	–	–	45.5
SQGE (Ours)	**38.5**	**49.5**	**31.2**	**40.4**	**46.4**
Improved	+8.5*	+1.3*	+5.9*	+5.6*	+0.9*

Table 6
The F1 scores of the compared model on the FewNYT dataset. **Bold** indicates the best results, and underlined indicates the second-best results.

Model	5W1S	5W5S	10W1S	10W5S	10W10S
Proto (Snell et al., 2017)	20.6	30.8	13.5	22.5	–
PTN (Fei et al., 2022)	33.3	42.0	26.7	34.2	–
RelATE (Cong et al., 2022)	27.1	46.0	19.4	35.8	38.9
SQGE (Ours)	**38.2**	**48.1**	**30.1**	**39.1**	**43.4**
Improved	+11.1*	+2.1*	+10.7*	+3.3*	+4.5*

Table 7
The F1 scores of the compared model on the TACRED dataset. **Bold** indicates the best results, and underlined indicates the second-best results.

Model	5W1S	5W5S	3W1S	3W5S
MatchNet (Vinyals et al., 2016)	11.8	17.7	19.9	23.8
Proto (Snell et al., 2017)	13.2	19.2	20.6	25.3
MLMAN (Ye & Ling, 2019)	18.4	25.0	27.4	28.9
TGIN (Wang et al., 2022)	20.4	27.7	30.4	31.2
SQGE (Ours)	**36.6**	**42.4**	**39.1**	**43.2**
Improved	+16.2*	+14.7*	+8.7*	+12.0*

It encodes the relation within the entity label and immediately extracts the head and tail entities. This approach enhances the complexity of the entity extraction process while also preventing the propagation of errors. Therefore, our relation triplet extraction results are similar to entity recognition results. PTN and RelATE employ a two-stage extraction scheme for other baselines, wherein relation extraction is carried out first, followed by entity recognition. This decomposed extraction scheme divides the relational triplet extraction, and each task is relatively independent. Therefore, the performance of a single task will be better, but the model has an error propagation problem, which affects the final relational triple extraction performance.

We selected some recent baselines to analyze the time complexity of the model, as shown in Table 9. We divide the time complexity of the model into two parts for calculation, namely the encoding part and the prototype construction part. The encoding part mainly calculates the time complexity generated when encoding each model. The time complexity of the prototype construction part refers to the time complexity generated by the model to build entity or relation prototypes, which is usually related to the number of relation categories and the prototype construction method. SQGE has an advantage in the time complexity of the prototype calculation part, mainly since SQGE uses a unified prototype construction method and does not need to divide entity and relation prototypes, because it can greatly reduce the time complexity. In the encoding calculation part, the time complexity of SQGE mainly depends on the size of k, which is consistent with our conclusion in Table 10. The spatial complexity of the model is mainly determined by the encoding calculation.

To further analyze the working efficiency of our model, we counted the training time, inference time, and memory occupation of the model under 5-way 1-shot and 5-way 5-shot conditions in Table 10 and compared them with the baseline model RelATE. There are two main reasons for choosing this baseline. First, as an open-source baseline, RelATE can be easily used for comparative experiments. Second, RelATE uses the relational decomposition method for FS-RTE, showing high efficiency. All experiments were conducted under the same settings as the baseline, including 40,000 training episodes and 3000 inference episodes, and the device was a 3090 GPU. SQGE has a large difference in time and space overhead, especially in the 5-way 5-shot scenario. Through specific ablation experiments, we found that the performance limitations of SQGE mainly stem from the entity-level feature enhancement module. This module increases the number of samples under each category, resulting in an increase in model performance overhead, which is difficult to avoid. However, we can achieve a balance between performance and overhead by adjusting the number of boosting samples, which is set to 5 for the current experiment. When this module is removed, the training time and memory occupation of SQGE are comparable to the baseline RelATE, and the inference time is significantly reduced, while the F1-score of the model is still better than the baseline. This shows that other modules of SQGE are still highly competitive and can effectively improve the performance of the model while maintaining model efficiency.

5.4. Parameters sensitivity analysis

We provide a comprehensive explanation of the model parameter settings in Section 5.2. However, we determine the selection through experimental results for some crucial hyperparameters. Fig. 5(a) shows that we train and test the model by setting different mask thresholds. When the mask threshold is set to 1, the support-query prototype guidance module is not used. The lower the mask threshold is set, the higher the model recall, the more correct instances the prototype guidance module can use, and the more incorrect samples it produces, which needs to be balanced. The model exhibits its lowest efficacy when the mask threshold is set to 0.9, and eliminating the module could be more advantageous. We analyze that the reason may be that the mask threshold is set too high, which excludes many correct samples, leaving less knowledge available to the model and thus producing negative effects. As shown in Fig. 5(b), we conducted experiments under different parameter settings and in Eq. (8), which denotes the proportion of support set prototypes and the proportion of query set prototypes. The experiments show that the best results are achieved when using prototypes that are a fusion of complete support set and query set prototypes.

In addition, we also analyzed and experimented with the parameters in the entity-level feature enhancement module, as shown in Table 11. By setting different numbers of enhanced entity and observing their corresponding F1 score, training time, and memory occupation, we selected the most appropriate parameter values. As the number of entities in the entity-level feature enhancement module increases, the model's training time, memory occupation, and F1 score also increase synchronously. When k is set to 5, a relatively balanced result can be obtained. When k is greater than 5, the performance of the model increases very slowly, but the training time and memory consumption still maintain a high growth rate. This also shows that when the number of entities reaches a certain level, the prototype can be constructed relatively well, and continuing to add entities will not provide more help. This conclusion is consistent with the conclusion obtained in the ablation experiment.

Table 8
Comparison of model F1 scores for entities, relations, and triples on the FewRel dataset. **Bold** denotes the best result, whereas underlined represents the second-best result.

Model	5-way 1-shot			5-way 5-shot			10-way 1-shot			10-way 10-shot		
	Relation	Entity	Triple	Relation	Entity	Triple	Relation	Entity	Triple	Relation	Entity	Triple
FS-GNN (Garcia & Bruna, 2018)	78.4	21.6	17.8	88.4	26.0	24.5	66.9	15.7	11.4	77.7	20.5	16.1
MatchNet (Vinyals et al., 2016)	75.8	18.7	15.4	84.6	20.7	18.7	59.4	12.5	8.2	77.8	20.0	16.3
MPE (Yu, Zhang et al., 2020)	–	–	–	**93.8**	25.0	23.3	–	–	–	**84.6**	14.9	12.1
Proto (Snell et al., 2017)	77.6	19.4	15.9	87.4	25.1	21.2	65.7	14.5	10.4	76.0	19.8	15.4
MLMAN (Ye & Ling, 2019)	82.5	23.4	20.4	91.8	30.5	28.5	70.7	20.4	15.4	81.9	23.3	19.2
TGIN (Wang et al., 2022)	**83.7**	27.5	24.0	93.1	33.6	32.3	**72.3**	22.8	17.3	83.7	26.6	22.8
NNM (He et al., 2022)	–	–	–	88.7	32.6	32.2	–	–	–	75.1	26.6	25.0
RelATE (Cong et al., 2022)	66.3	46.5	28.7	79.9	**59.6**	42.3	54.7	37.9	20.3	75.4	**57.0**	40.9
PTN (Fei et al., 2022)	81.1	**47.2**	30.0	84.2	56.9	40.0	68.7	**39.4**	25.3	77.6	52.4	36.2
SQGE (Ours)	69.5	38.9	**38.5**	80.1	50.2	**49.5**	58.3	31.8	**31.2**	76.5	47.2	**46.4**
Improved	−14.2	−8.3	+8.5*	−13.0	−9.4	+7.2*	−14.0	−7.6	+5.9*	−8.1	−9.8	+5.5*

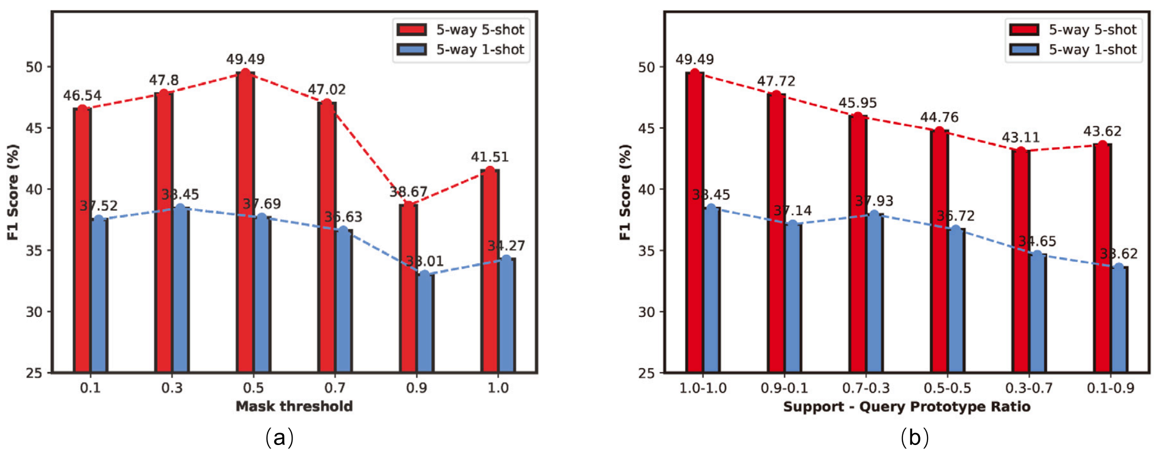

Fig. 5. Experimental results of the model under (a) different mask threshold settings and (b) support-query prototype ratio setting.

Table 9
Comparison and analysis of the time complexity of the models, where n represents the sentence length, k represents the number of entities in the entity-level feature enhancement module, and N represents the number of relation categories. For example, in 5-way 5-shot, N is 5.

Model	Encode computing	Prototype computing	All
TGIN (Wang et al., 2022)	$2n$	$2N^2$	$2(N^2+2)$
RelATE (Cong et al., 2022)	$2n$	$N+4N+1$	$2n+5N+1$
PTN (Fei et al., 2022)	n	N^2+4N+1	$n+N^2+4N+1$
TLSM (Jiang et al., 2023)	n	$N+4N+1$	$n+5N+1$
RCTE (Liao et al., 2024)	$3n$	$N+4N+1$	$3n+5N+1$
SQGE (Ours)	kn	$2N+1$	$kn+2N+1$

5.5. Ablation studies

We performed ablation studies to examine the effects of each distinct component in our model. From Table 12, where removing each component of the model results in performance degradation of relational triplet extraction. The entity-level feature enhancement strategy can significantly improve the F1 value of the model from 26.38% to 31.78% under 5-way 1-shot, representing a 5.4% increase. In the 5-way 5-shot, the entity-level feature enhancement strategy only increased the F1 value of the model by 1.29%, while the support-query prototype guidance strategy achieved the best performance improvement, from 33.92% to 37.49%, 3.57% improvement. This is because the sample volume is very scarce in 1-shot settings, and entity character enhancement strategies are more helpful for the model. The entity-level feature enhancement strategy can enhance the representation of the entity prototype by integrating more information about similar entities. When the number of entities is 1, in the 1-shot setting, a single entity represents the entity prototype, which often produces a large deviation. In the 5-shot setting, as the number of entities increases, the representation of the entity prototype tends to be more perfect. Therefore, the entity-level feature enhancement strategy will play a greater role in the scenario with fewer samples. The support-query prototype guidance strategy can play a greater role when the base model is better, which is also confirmed by subsequent analysis. For the support-query prototype guidance strategy, the F1 value of the model increased from 26.38% to 29.71% under 5-way 1-shot, 3.33% improvement. However, as the base model improves, the effect of the support-query prototype guidance strategy becomes increasingly better. For example, in the 5-way 1-shot, based on the use of both entity-level feature enhancement and multi-level contrastive learning strategies, the support-query prototype guidance strategy increased the F1 value of the model from 34.37% to 38.45%, an increase of 4.08%. Especially under the 5-way 5-shot setting, the prototype guidance strategy increased the model's F1 value from 40.55% to 49.49%, an increase of 8.94%. The support-query prototype guidance strategy is mainly divided into two steps. First, the query data mask is obtained based on the support set prototype to build the query set prototype. Then the query set prototype is fused with the support set prototype to obtain the final prototype to predict the relational triples in the sentence. Therefore, when there is a stronger basic model, the initial matching results of the model will be more accurate, resulting in a better query set prototype and better model performance.

Furthermore, based on the results of the ablation experiments, we found that there is a mutually reinforcing effect between the various proposed components. Specifically, the performance improvement brought by the combination of each component is significantly higher than the effect of using a component alone. For example, in the multi-level contrastive learning strategy, when only using this strategy, the performance of the model under the 5-way 1-shot and 5-way 5-shot tasks is improved by 2.45% and 1.45%, respectively. However, when

Table 10
Efficiency analysis of the model. We selected the specific experimental data of RelATE and SQGE on the dataset FewRel for analysis. MO: Memory Occupation (G), TT: Training Time (m), IT: Inference Time (s). "ELFE" denotes the entity-level feature enhancement, "MLCL" denotes the multi-level contrastive learning, and "SQPG" denotes the support-query prototype guidance.

Setting	5-way 1-shot				5-way 5-shot			
	TT(m)	IT(s)	MO(G)	F1-score	TT(m)	IT(s)	MO(G)	F1-score
RelATE (Cong et al., 2022)	275	394	4.65	28.7	452	585	6.91	42.3
SQGE	391	291	6.79	38.5	1043	1116	14.89	49.5
w/o ELFE	268	152	5.15	34.51	382	292	7.01	43.36
w/o SQPG	233	131	4.85	28.83	345	274	6.51	35.37
w/o MLCL	221	118	4.82	26.38	331	245	6.49	33.92

Table 11
Hyperparameter settings in the entity-level feature enhancement module are mainly performed on the FewRel dataset. k represents the number of enhanced entities, MO: Memory Occupation (G), TT: Training Time (m).

Setting	5-way 1-shot				5-way 5-shot			
	k = 1	k = 3	k = 5	k = 7	k = 1	k = 3	k = 5	k = 7
TT(m)	268	324	391	432	382	619	1043	1247
MO(G)	5.15	6.21	6.79	8.02	7.01	11.4	14.89	18.7
F1	34.51	36.98	38.45	38.94	43.36	45.96	49.49	48.46

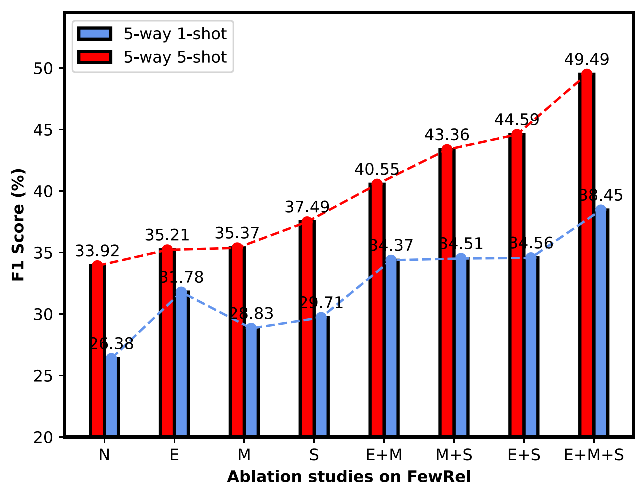

Fig. 6. The performance change trend of each component. 'N' denotes no components, 'E' denotes ELFE, 'M' denotes MLCL, 'S' denotes SQPG.

this strategy is combined with entity feature enhancement and support query prototype guidance strategies, the model improvement increases significantly to 3.89% and 4.9%. The reason for this improvement is that when the entity category representation is more accurate, the model can construct more distinctive positive and negative samples, which helps to learn the characteristics of positive and negative samples more effectively. In addition, this also proves that our model focuses on the overall intra-class and inter-class gaps, and the various components can cooperate effectively without redundancy. In the ablation experiment, the performance change trend of each component is shown in Fig. 6.

5.6. Prototype visualization and analysis

5.6.1. Multi-level comparative learning

To visualize the effect of multi-level comparative learning on the model, we randomly selected six entity categories from the test set, and the results are shown in Fig. 7. (1) Under the setting of 5-way 1-shot, compared with Figs. 7(a), 7(b) has apparent separation between prototypes due to the use of multi-level contrastive learning. In contrast, the prototype of Fig. 7(a) has a mixture of many different

Table 12
Ablation studies on FewRel. "ELFE" denotes the entity-level feature enhancement, "MLCL" denotes the multi-level contrastive learning and "SQPG" denotes the support-query prototype guidance.

ELFE	MLCL	SQPG	5-way 1-shot	5-way 5-shot
			26.38	33.92
✓			31.78 $_{\uparrow 5.40}$	35.21 $_{\uparrow 1.29}$
	✓		28.83 $_{\uparrow 2.45}$	35.37 $_{\uparrow 1.45}$
		✓	29.71 $_{\uparrow 3.33}$	37.49 $_{\uparrow 3.57}$
✓	✓		34.37 $_{\uparrow 7.99}$	40.55 $_{\uparrow 6.63}$
	✓	✓	34.51 $_{\uparrow 8.13}$	43.36 $_{\uparrow 9.44}$
✓		✓	34.56 $_{\uparrow 8.18}$	44.59 $_{\uparrow 10.67}$
✓	✓	✓	**38.45** $_{\uparrow 12.07}$	**49.49** $_{\uparrow 15.57}$

categories. This demonstrates that our contrastive learning strategy effectively separates different types of samples. Minimizing the spatial distance among samples of the same classes and increasing the spatial distance among samples of other classes reduces the inter-class gaps. (2) Under the 5-way 5-shot setting, compared with Fig. 7(c), the prototypes in Fig. 7(d) are closer together, and the distance between them is relatively balanced, neither too close nor too far, which shows that our contrastive learning strategy can effectively cluster samples of the same type. (3) Under the 5-way 5-shot setting, the visualization effect of the prototype is much better than that of the 5-way 1-shot, which shows that increasing the number of samples can significantly improve the learning effect of the model. (4) Comparing Figs. 7(c) and 7(d), we can see that prototypes of the same type are not always grouped in the same cluster, as there can often be more detailed classifications under the category. In the future, we can try to explore multi-prototype solutions.

5.6.2. Entity-level feature enhancement

To intuitively feel the impact of enhancing entity features on the model, we selected the same six categories from the test set used in the multi-level contrastive learning prototype visualization. As shown in Fig. 8. (1) Fig. 8(a) displays numerous category-confusing prototypes due to the variability among individual samples in the 5-way 1-shot. Conversely, Fig. 8(b) exhibits a notable enhancement in the distribution of prototypes due to the implementation of entity augmentation, which improves the representation of prototypes belonging to the same class of entity. (2) Upon comparing Figs. 8(a) and 8(c), it is evident that in the 5-way 5-shot scenario, the prototype distribution becomes more rational as the base sample size increases. However, it is worth noting that specific prototypes still exhibit category crossovers. (3) The number of prototype cross-samples in Fig. 8(d) is lower than in Fig. 8(c), indicating that the entity enhancement technique remains effective under the 5-way 5-shot setting. However, the effectiveness steadily diminishes as the base sample size increases. The entity-level feature enhancement strategy effectively strengthens the feature representation of samples in the same classes, reducing the inter-class gaps.

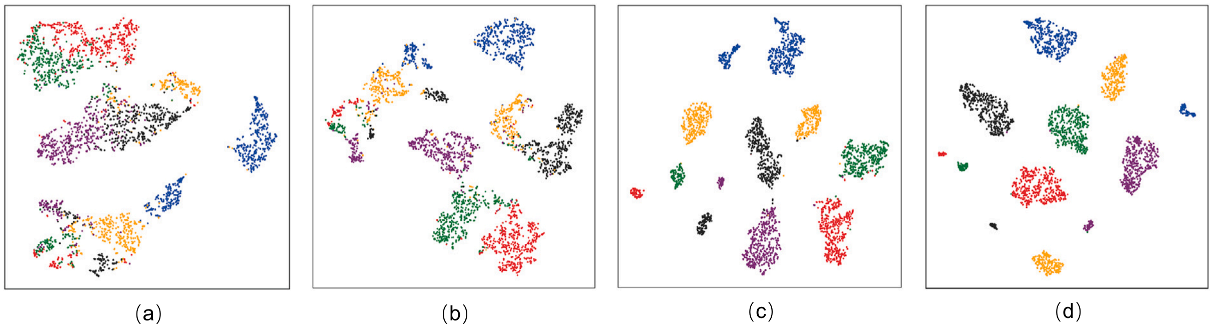

Fig. 7. Visualization of MLCL prototypes. (a) Remove the MLCL module under 5-way 1-shot. (b) SQGE under 5-way 1-shot. (c) Remove the MLCL module under 5-way 5-shot. (d) SQGE under the 5-way 5-shot.

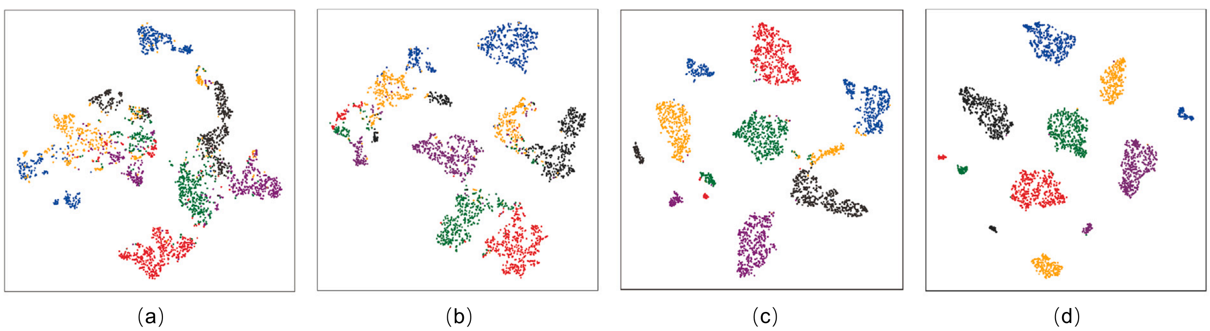

Fig. 8. Visualization of Entity-Level Feature Enhanced prototypes. (a) Remove the ELFE module under 5-way 1-shot. (b) SQGE under 5-way 1-shot. (c) Remove the ELFE module under 5-way 5-shot. (d) SQGE under 5-way 5-shot.

5.6.3. Support-query prototype guidance

We performed a similarity comparison of prototypes between different datasets under 5-way 5-shot. The comparison is performed by calculating the similarity between the prototypes and the golden labels of the query set in each test set episode and then averaging all the results. It is evident that the higher the degree of similarity between them, the more accurate the recognition result. Fig. 9. shows that the similarity between the hybrid prototype built based on the support-query prototype guidance method and the query set is the highest, around 0.82. It demonstrates that based on the existing information in the support set, a more complete prototype can be constructed by mining the characteristics of the query set. The similarity between the support set prototype and the query set is the lowest, which shows that it is not enough to rely solely on the data in the support set to construct a prototype. There is an intra-class deviation between the support set and the query set. The similarity between the query set prototype and the query set is about 0.75. The support set can mine helpful features in the query set. However, due to the existence of errors, the support set is still needed as an additional feature to improve the prototype. The support-query prototype guidance strategy can close the intra-class gap, resulting in a more accurate prototype and an improved recognition effect for the query set.

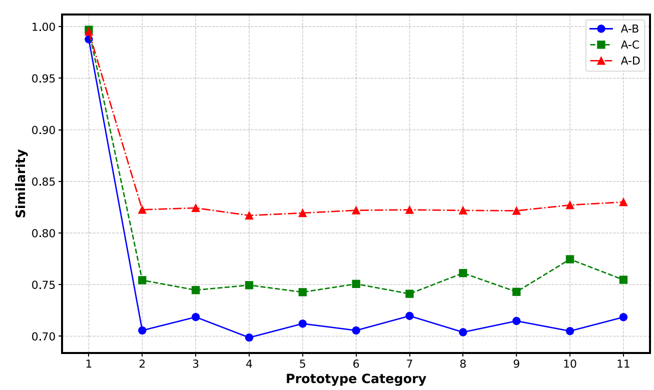

Fig. 9. Prototype similarity comparison between different data in the 5-way 5-shot. Where ‘A’ represents the entity representation in the query set, ‘B’ represents the prototype built based on the support set, ‘C’ represents the prototype built based on the query set, and ‘D’ represents the hybrid prototype built based on support-query guidance. The horizontal coordinates indicate the type of prototype, where ‘1’ denotes prototype ‘O’ and the others are head entity and tail entity prototypes with different relations.

5.7. Case study

Through case study, we effectively demonstrated the performance of the SQGE model and deeply analyzed the errors and their causes in the extraction process. The experimental results are shown in Fig. 10, which clearly reflects the performance of the model in practical applications. We summarize the following five main types of errors:

- Entity errors. Although the model can correctly identify the relation, it extracts the wrong entity, which may lead to misleading or incomplete information in practical applications.
- Boundary errors. Although the model can correctly identify the relation, it extracts the wrong entity, which may lead to misleading or incomplete information in practical applications.
- Relation errors. In this case, the model correctly identified the entities but incorrectly classified the relations between them. This type of error usually occurs when the similarity between relation types is high, making it difficult for the model to effectively distinguish them. This also reflects the complexity of relation extraction, especially in judging between similar relations. However, among all recognition error samples, the number of entity errors is much higher than the proportion of relation errors, which shows that entity recognition is more complicated.

Instance 1: In addition to its sister model the Sprinter there was a redesigned - body version built by Toyota affiliate Daihatsu called the Daihatsu Charmant.			
SQGE	**True**: <Daihatsu Charmant, manufacturer, Toyota>	**Pred**: <Charmant, manufacturer, Daihatsu>	①
Instance 2: Alien : Covenant is a soundtrack album for the 2017 film composed by Jed Kurzel.			
SQGE	**True**: <Alien : Covenant, composer, Jed Kurzel>	**Pred**: <Covenant, composer, Jed Kurzel>	①②
Instance 3: Deixa Ele Sofrer debuted at forty - eight on the Brazil Billboard Hot 100 Airplay.			
SQGE	**True**: <Hot 100 Airplay, publisher, Brazil>	**Pred**: <Hot 100 Airplay, manufacturer, Billboard>	①③
Instance 4: Use byobu for extended features in your terminal window ghacks.net Byobu 3.0 reworked the build system to use automake and allow for porting to other Unix - like operating systems.			
SQGE	**True**: <automake, operating system, Unix-like>	**Pred**: <automake, operating system, Unix>	①④
Instance 5: Sports champions is a 2010 sports video game developed by san diego studio and zindagi games and published by sony computer entertainment for playstation 3 which utilizes playstation move .			
SQGE	**True**: <playstation 3, manufacturer, sony computer entertainment>	**Pred**: <sports champions, publisher, zindagi games>	①②⑤
Instance 6: In a bid to improve wet weather braking performance Honda fitted inboard brakes to models such as the VF400F and CBX500F.			
SQGE	**True**: <VF400F, manufacturer, Honda>	**Pred**: <CBX500F, manufacturer, Honda>	①⑤
Instance 7: Quest publishes original articles written in either English or French each with a summary in the other language.			
SQGE	**True**: <Quest, original language of film or TV show, English>	**Pred**: <Quest, original language of film or TV show, French>	①⑤

Fig. 10. Common scenarios for errors of relational triple on FewRel dataset under 5-way 1-shot. Different colors represent different entities and relations: green for head entities, blue for tail entities, red for misidentified entities, and orange for relations. Different serial numbers indicate the type of error, where ① indicates entity error, ② indicates boundary error, ③ indicates relation error, ④ indicates data annotation error, and ⑤ indicates multi-triplet problem. It is important to note that a sentence can have multiple error types.

Instance 1: After moving to London he began to write for "NME" initially under the name Susan Williams.			
True	<Susan Williams, work location, London> ✓	**SQGE**	<Susan Williams, work location, London> ✓
Baseline	<Susan Williams, work location, London> ✓	**RelATE**	<Susan Williams, work location, London> ✓
Instance 2: PowerDirector runs on Windows 7 through Windows 10 with 64-bit versions recommended.			
True	<PowerDirector, operating system, Windows 10> ✓	**SQGE**	<PowerDirector, operating system, Windows 10> ✓
Baseline	<PowerDirector, operating system, Windows 7> ×	**RelATE**	<Power, operating system, Windows> ×
Instance 3: Alongside the DKW F89 passenger car it was the first vehicle to be manufactured by the new Auto Union conglomerate in Ingolstadt following the reestablishment of the business in West Germany.			
True	<DKW F89, manufacturer, Auto Union> ✓	**SQGE**	<DKW F89, manufacturer, Auto Union> ✓
Baseline	<DKW, applies to jurisdiction, Germany> ×	**RelATE**	<DKW F89 … new, manufacturer, Auto> ×
Instance 4: Penzance was the birthplace of Maria Branwell mother of three famous novelists – Charlotte Bronte, Emily Bronte and Anne Bronte.			
True	<Anne Bronte, sibling, Emily Bronte> ✓	**SQGE**	<Emily Bronte, sibling, Maria Branwell> ×
Baseline	<Empty Triple> ×	**RelATE**	<Empty Triple> ×

Fig. 11. Instances of different models on the FewRel dataset under 5-way 1-shot. Different colors represent different entities and relations: green for head entities, blue for tail entities, red for misidentified entities, and orange for relations.

- Labeling errors. Some samples in the dataset are inaccurately labeled, but the model's recognition results are correct, which shows the important impact of data quality on model performance.
- Multi-triplet problem. Although the triplets identified by our model are not completely consistent with the golden annotations, they are all reasonable. This situation reflects that there may be multiple correct triplets in the dataset, and not all triplets have been labeled. This shows that when the dataset is constructed, more triplets can be introduced to enhance the learning ability of the model.

These error types not only reveal the current limitations of the model but also provide directions for future improvements. By optimizing these specific problems, the performance of the SQGE model in entity relationship extraction tasks can be further improved.

To display the advantages of SQGE more intuitively, we selected the following two models for case analysis. (1) Baseline, the SQGE model that removes entity-level feature enhancement, multi-level contrastive learning, and support-query prototype guidance module. (2) RelATE, a model for FS-RTE based on relational decomposition, is the best baseline model among open-source code. Fig. 11 shows four typical qualitative results, which are as follows. First, all models get correct triples for simple entities and relations, such as person and place, in instance 1. Second, RelATE annotates the start/end position of the entity compared with SQGE, which is more likely to produce incorrect entity boundaries and makes it challenging to capture entity semantic information entirely. For example, RelATE only recognizes part of the head and tail entities in the second instance. In the third instance, RelATE misidentifies the span of the header entity. Third, the proposed modules of entity-level feature enhancement, multi-level contrastive learning, and support-query prototype guidance can effectively enhance the model to solve the problem of intra-class gap and inter-class gap, then improve the model effect. In instance 3, due to the complex sentence structure, the Baseline model had difficulty identifying the triples of the sentence and even misidentifying the relations, while SQGE can accurately identify them. Finally, although all models did not accurately identify triples that require sentence reasoning, SQGE still showed certain advantages. In instance 4, SQGE correctly identifies relations and head entities, whereas Baseline and RelATE can only extract empty triples. Note that the order of the head and tail entities can be reversed for the' sibling' relation.

6. Conclusion

In this paper, we propose an FS-RTE method based on support-query prototype guidance and enhancement, which can effectively alleviate the intra-class and inter-class gaps. We enhance the representation of the same class of target entities through entity characteristics and perform supervised contrast learning between support and query sets separately. Thus, while strengthening the exact target class representation, we separate the characteristic space of the target class from the other classes, reducing the inter-class gap. Furthermore, we developed a support-query prototype guidance module for constructing a query prototype using the support set prototype. We then used these two prototypes to detect relational triples within sentences. The integrated prototype can accurately capture the consistent fundamental characteristics of the query object, correctly align the query features, and minimize the intra-class gap. Experimental results show that our proposed method can effectively alleviate intra-class and inter-class gaps, improve prototype quality, and achieve state-of-the-art performance. However, SQGE still has some limitations. We conducted a detailed error analysis in the case study to describe the strengths and weaknesses of the model. For example, SQGE has particular difficulties in processing multiple triples in a sentence and faces problems in computational performance and efficiency problems when processing more relation categories. In the future, we will explore FS-RTE in multi-triple scenarios. Multi-triple extraction will face more complex situations, with more significant interaction and overlap among triples, posing greater challenges.

CRediT authorship contribution statement

Chen Gao: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology. **Xuan Zhang:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization. **Zhi Jin:** Writing – review & editing, Conceptualization. **Wei Cai:** Visualization. **Danyang Wang:** Data curation. **Kunpeng Du:** Visualization, Data curation. **Chunlin Yin:** Formal analysis. **Tong Li:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Xingdian Talent Support Program Industrial Innovation Talent Project of Yunnan under Grant No. yfgrc202422; Science Foundation of Young and Middle-aged Academic and Technical Leaders of Yunnan; Science Foundation of Yunnan Jinzhi Expert Workstation; Major Project of Yunnan Natural Science Foundation; Special funding projects for the national guidance of provincial science and technology development; Knowledge-driven Smart Energy Science and Technology Innovation Team of Yunnan Provincial Department of Education; Open Foundation of Yunnan Key Laboratory of Software Engineering; Yunnan University Graduate Research Innovation Foundation.

Data availability

Data will be made available on request.

References

Chen, J., Liu, Q., Lin, H., Han, X., & Sun, L. (2022). Few-shot named entity recognition with self-describing networks. arXiv preprint arXiv:2203.12252.

Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, *4*, 357–370.

Cong, X., Sheng, J., Cui, S., Yu, B., Liu, T., & Wang, B. (2022). Relation-guided few-shot relational triple extraction. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 2206–2213).

Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia: Case studies on organization and retrieval* (pp. 21–49). Springer.

Dang, Z., Deng, C., Yang, X., Wei, K., & Huang, H. (2021). Nearest neighbor matching for deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13693–13702).

Das, S. S. S., Katiyar, A., Passonneau, R. J., & Zhang, R. (2022). Container: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 6338–6353).

Dong, B., Yao, Y., Xie, R., Gao, T., Han, X., Liu, Z., et al. (2020). Meta-information guided meta-learning for few-shot relation classification. In *Proceedings of the 28th international conference on computational linguistics* (pp. 1594–1605).

Fei, J., Zeng, W., Zhao, X., Li, X., & Xiao, W. (2022). Few-shot relational triple extraction with perspective transfer network. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 488–498).

Feng, J., Xu, G., Wang, Q., Yang, Y., & Huang, L. (2024). Note the hierarchy: Taxonomy-guided prototype for few-shot named entity recognition. *Information Processing & Management*, *61*(1), Article 103557.

Fritzler, A., Logacheva, V., & Kretov, M. (2019). Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing* (pp. 993–1000).

Gao, T., Han, X., Liu, Z., & Sun, M. (2019). Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 6407–6414).

Gao, C., Zhang, X., Jin, Z., Shang, W., Ma, Y., Li, L., et al. (2024). Few-shot relational triple extraction with hierarchical prototype optimization. *Pattern Recognition*, *156*, Article 110779.

Garcia, V., & Bruna, J. (2018). Few-shot learning with graph neural networks. In *6th international conference on learning representations, ICLR 2018* (pp. 1–10).

Han, J., Cheng, B., Wan, Z., & Lu, W. (2023). Towards hard few-shot relation classification. *IEEE Transactions on Knowledge and Data Engineering*.

Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., et al. (2018). FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4803–4809).

He, X., Song, H., Cheng, D., & Xu, B. (2022). Few-shot relational triple extraction with nearest neighbor matching. In *International conference on computer graphics, artificial intelligence, and data processing*: *vol. 12168*, (pp. 262–266). SPIE.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., et al. (2021). Knowledge graphs. *ACM Computing Surveys*, *54*(4), 1–37.

Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2021). Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5149–5169.

Huang, J., Li, C., Subudhi, K., Jose, D., Balakrishnan, S., Chen, W., et al. (2021). Few-shot named entity recognition: An empirical baseline study. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 10408–10423).

Ji, B., Li, S., Gan, S., Yu, J., Ma, J., & Liu, H. (2022). Few-shot named entity recognition with entity-level prototypical network enhanced by dispersedly distributed prototypes. arXiv preprint arXiv:2208.08023.

Jiang, S., Zhu, J., & He, L. (2023). Few-shot relational triple extraction based on evaluation of token-level semantic similarity. In *International conference on artificial neural networks* (pp. 232–242). Springer.

Kenton, J. D. M. -W. C., & Toutanova, L. K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT* (pp. 4171–4186).

Kingma, D. (2014). Adam: A method for stochastic optimization. In *Int conf learn represent* (pp. 1–10).

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). 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* (pp. 260–270).

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551.

Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, *34*(1), 50–70.

Liao, B., Lu, Z., & Guo, Y. (2024). RCTE: Relation candidate-guided few-shot relational triple extraction. In *Proceedings of the 2024 3rd international conference on cyber security, artificial intelligence and digital economy* (pp. 486–491).

Ma, J., Ballesteros, M., Doss, S., Anubhai, R., Mallya, S., Al-Onaizan, Y., et al. (2022). Label semantics for few shot named entity recognition. arXiv preprint arXiv:2203.08985.

Mansouri, A., Affendey, L. S., & Mamat, A. (2008). Named entity recognition approaches. *International Journal of Computer Science and Network Security*, *8*(2), 339–344.

Ning, J., Yang, Z., Sun, Y., Wang, Z., & Lin, H. (2023). Od-rte: A one-stage object detection framework for relational triple extraction. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 11120–11135).

Riedel, S., Yao, L., & McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 148–163).

Sarawagi, S., et al. (2008). Information extraction. *Foundations and Trends® in Databases*, *1*(3), 261–377.

Shang, Y. -M., Huang, H., & Mao, X. (2022). Onerel: Joint entity and relation extraction with one module in one step. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 36*, (10), (pp. 11285–11293).

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, *30*.

Sun, S., Sun, Q., Zhou, K., & Lv, T. (2019). Hierarchical attention prototypical networks for few-shot text classification. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 476–485).

Tan, Z., Zhao, X., Wang, W., & Xiao, W. (2019). Jointly extracting multiple triplets with multilayer translation constraints. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 33*, (01), (pp. 7080–7087).

Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J. B., & Isola, P. (2020). Rethinking few-shot image classification: A good embedding is all you need? In *Computer vision–ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, part XIV 16* (pp. 266–282). Springer.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, *29*.

Wang, L., Cao, Z., De Melo, G., & Liu, Z. (2016). Relation classification via multi-level attention cnns. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1298–1307).

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, *53*(3), 1–34.

Wang, Y., Yu, B., Zhang, Y., Liu, T., Zhu, H., & Sun, L. (2020). Tplinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th international conference on computational linguistics* (pp. 1572–1582).

Wang, J., Zhang, L., Liu, J., Ma, K., Wu, W., Zhao, X., et al. (2022). TGIN: Translation-based graph inference network for few-shot relational triplet extraction. *IEEE Transactions on Neural Networks and Learning Systems*.

Wei, Z., Su, J., Wang, Y., Tian, Y., & Chang, Y. (2020). A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1476–1488).

Wen, M., Xia, T., Liao, B., & Tian, Y. (2023). Few-shot relation classification using clustering-based prototype modification. *Knowledge-Based Systems*, *268*, Article 110477.

Xiao, Y., Jin, Y., & Hao, K. (2021). Adaptive prototypical networks with label words and joint representation learning for few-shot relation classification. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(3), 1406–1417.

Ye, Z. -X., & Ling, Z. -H. (2019). Multi-level matching and aggregation network for few-shot relation classification. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 2872–2881).

Yu, Y., Ji, Z., Han, J., & Zhang, Z. (2020). Episode-based prototype generating network for zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14035–14044).

Yu, H., Zhang, N., Deng, S., Ye, H., Zhang, W., & Chen, H. (2020). Bridging text and knowledge with multi-prototype embedding for few-shot relational triple extraction. In *Proceedings of the 28th international conference on computational linguistics* (pp. 6399–6410).

Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers* (pp. 2335–2344).

Zhang, Y., Zhong, V., Chen, D., Angeli, G., & Manning, C. D. (2017). Position-aware attention and supervised data improve slot filling. In *Conference on empirical methods in natural language processing* (pp. 1–10).

Zhao, T., Yan, Z., Cao, Y., & Li, Z. (2021). A unified multi-task learning framework for joint extraction of entities and relations. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 35*, (16), (pp. 14524–14531).

Zheng, H., Wen, R., Chen, X., Yang, Y., Zhang, Y., Zhang, Z., et al. (2021). PRGC: Potential relation and global correspondence based joint relational triple extraction. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing* (pp. 6225–6235).

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., et al. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 207–212).

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., et al. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76.