Enhancing Discriminative Representation in Similar Relation Clusters for Few-Shot Continual Relation Extraction

Duc Anh Le[1]*, Nam Le Hai[1]*, Thanh Xuan Nguyen[2]*, Linh Ngo Van[1], Diep Thi-Ngoc Nguyen[3], Sang Dinh[1], Thien Huu Nguyen[4]

[1]Hanoi University of Science and Technology, [2]Oraichain Labs Inc., US,
[3]VNU University of Engineering and Technology, [4]University of Oregon

Abstract

Few-shot Continual Relation Extraction (FCRE) has emerged as a significant challenge in information extraction, necessitating that relation extraction (RE) systems can sequentially identify new relations with limited labeled samples. While existing studies have demonstrated promising results in FCRE, they often overlook the issue of similar relations, which is a critical factor contributing to catastrophic forgetting. In this work, we propose SIRUS, a novel method that utilizes relation descriptions and dynamic clustering on these descriptions to identify similar relations. Leveraging this information, we introduce innovative loss functions specifically designed to enhance the distinction between relations, with a focus on learning to differentiate similar ones. Experimental results show that our approach can effectively mitigate the problem of catastrophic forgetting and outperforms state-of-the-art methods by a large margin. Additionally, we explore the potential of Large Language Model Embeddings (LLMEs) with representation learning and embedding capabilities, demonstrating their promise for advancing FCRE systems.

1 Introduction

Relation Extraction (RE) is a fundamental task in natural language processing (NLP) that involves recognizing relationships between entities from underlying content. Traditional relation extraction approaches commonly require substantial labeled datasets and assume a static collection of predefined relationships (Sun et al., 2020; Cabot and Navigli, 2021; Tang et al., 2022). In real-world settings, various specialized fields, such as scientific research (Kruiper et al., 2020), medicine (Luo et al., 2022), or law (Hendrycks et al., 2021), are rapidly advancing, leading to a continuous expansion in the diversity of relationships (Le et al., 2024c, 2025). Consequently, RE systems are required to exhibit adaptability to handle these evolving changes effectively. Besides, another challenge in developing RE models is the scarcity of annotated data for emerging relations.

To this end, the concept of Few-shot Continual Relation Extraction (FCRE) has been proposed (Qin and Joty, 2022; Chen et al., 2023) to enable the continuous learning of new relations from a limited number of samples. FCRE is a branch of continual information extraction (Le et al., 2025; Nguyen et al., 2023; Le et al., 2024b; Dao et al., 2024). However, due to the continual learning process with limited data, FCRE models are often biased toward the current task, facing the challenges of *overfitting* (Hawkins, 2004) and *Catastrophic forgetting* (Thrun and Mitchell, 1995; Le et al., 2024a; Hai et al., 2024; Van et al., 2022; Phan et al., 2022). Several methods have been introduced to address these issues, with memory-based approaches integrated with contrastive learning emerging as a prominent paradigm (Wang et al., 2023; Ma et al., 2024; Luo et al., 2024; Tran et al., 2024; Nguyen et al., 2025). These methods typically involve retaining a few representative samples from previous tasks and applying contrastive learning to ensure that the representations of samples across different relations remain sufficiently distinguishable.

However, none of the mentioned methods have considered the confusion between similar relations, which has been identified as a significant factor contributing to catastrophic forgetting in continual relation extraction (Wang et al., 2022; Zhao et al., 2023). This phenomenon becomes even more critical in the FCRE scenario, where the few available samples for each relation may not sufficiently represent the relations causing models to ignore subtle distinctions between similar relations, leading to catastrophic forgetting. A recent method, ConPL, introduced by Chen et al. 2023, addresses this is-

*Equally contributed.
†Corresponding author: namlh@soict.hust.edu.vn

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2450–2467

sue by identifying similar classes based on the distance between relation prototypes and examples, and employs focal loss to emphasize the distinctions between these similar classes. The relation prototypes are computed by averaging the representations of entities within the same relations; however, a limited number of samples may fail to produce sufficiently representative prototypes (Wang et al., 2023). Moreover, the problem of inconsistencies resulting from varying context sentences can render these prototypes unstable (Li and Lyu, 2024), reducing the effectiveness of this approach in identifying similar classes in the FCRE scenario.

Recent studies have leveraged label descriptions (Luo et al., 2024; Li and Lyu, 2024; Sainz et al., 2021; Borchert et al., 2024) for few-shot relation extraction, demonstrating their effectiveness in enriching representations and stabilizing label prototypes. Especially, Nguyen et al. (2025) utilizes LLMs to augment data by generating additional label descriptions and multiple samples for each relation type. However, in this work, we focus solely on exploiting the original relation descriptions without using an LLM-based data augmentation mechanism. We propose SIRUS, a novel method to enhance the Discriminative Representation in **Si**milar **R**elation Cl**us**ters, involves utilizing relation descriptions for label representation and employing dynamic clustering. As a result, the learning process enhances the differentiation of samples across relations, with a particular focus on similar relations, thereby reducing the phenomenon of catastrophic forgetting.

Furthermore, pre-trained Large Language Models (LLMs) with billions of parameters excel in autoregressive text generation tasks (Dubey et al., 2024; Jiang et al., 2023) and demonstrate strong performance on downstream tasks with only a few examples (Brown, 2020; Kojima et al., 2022), making them a promising approach for application in FCRE. They have also been explored in text classification and information extraction (Zhao et al., 2021; Wei et al., 2023); however, they often underperform compared to discriminative encoder models like BERT due to their generation-focused mechanism potentially makes them less effective for text representation learning. In the FCRE scenario, a recent study by Tran et al. (2024) has explored the capabilities of LLMs; however, it retained the use of causal language modeling (CLM) and applied a classification head to the last token, which may not fully exploit the embedding ability of LLMs. Several recent studies (BehnamGhader et al., 2024; Li et al., 2024; Lee et al., 2024) have investigated the capabilities of LLMs in representation learning by removing the causal mask and fine-tuning LLMs with contrastive learning, referred to as Large Language Model Embeddings (LLMEs), demonstrating promising results in retrieval and classification tasks. However, their ability for continual learning, particularly in the context of FCRE, remains unexplored. Therefore, we conduct comprehensive experiments on these LLMEs, offering valuable insights into the forgetting phenomenon within these models and their performance outcomes in FCRE settings.

In summary, our contributions are as follows:

1. We present a novel approach to address the issue of similar classes by utilizing relation description representation and subsequently employing dynamic clustering to identify groups of similar relations.

2. Leveraging information from similar relations, we propose three innovative loss functions to improve the distinction between samples from different relations. Ablation studies demonstrate the efficacy of each loss function.

3. We are the first to examine LLMEs with representation learning capabilities within the context of FCRE. Our findings indicate that these models continue to suffer from the issue of catastrophic forgetting. However, applying our methods significantly enhances their performance, surpassing both the use of BERT encoder backbones and original LLMs with causal language modeling.

4. Extensive experiments conducted on two FCRE benchmarks, TACRED and FewRel, demonstrate the effectiveness of our proposed framework and highlight the promising results achieved through the use of LLMEs.

2 Background

2.1 Problem Formulation

Few-Shot Continual Relation Extraction (FCRE) presents a challenging paradigm in natural language processing, combining the complexities of continual learning with the constraints of few-shot scenarios. Some related works are discussed in

Appendix A. In this framework, a model confronts a series of tasks $\mathcal{T} = \{\mathcal{T}^1, \mathcal{T}^2, ..., \mathcal{T}^n\}$, each introducing a set of novel relations R_i to be learned. For every task $\mathcal{T}^i$, the model is provided with a limited dataset $\mathcal{D}_i = \{(x_j, r_j)\}_{j=1}^m$, where $m = N \times K$ represents the total number of examples, where N represents the number of new relations and K denotes the few-shot sample size for each relation. Each example consists of an input sentence x_j containing a pair of entities (e_h, e_t), and a corresponding relation label $y_j \in R_i$. This task configuration is also known as *"N-way-K-shot"* learning setting, as introduced by Chen et al. (2023). Finally, evaluation is conducted on a comprehensive test set $\mathcal{D}^{test}$ that contains all relations $R_{total} = \bigcup_{i=1}^n R_i$ encountered across tasks, assessing both its ability to learn new relations and retain proficiency in earlier ones. This formulation encapsulates the essence of FCRE, highlighting its significance in developing adaptive and efficient relation extraction systems.

The core challenge of FCRE is twofold: the model must quickly adapt to new relations with limited examples (few-shot learning) while maintaining knowledge of previously learned relations (continual learning). This requires balancing the model's plasticity for learning new relations and its stability for preserving prior knowledge.

2.2 Input Formulation and Representation

In Relation Extraction, the foundational deep learning approach (Ji et al., 2020; Wang and Lu, 2020) typically involves encoding input data with a pretrained language model (PLM) like BERT (Devlin et al., 2019). A crucial aspect of RE is how to formulate the input effectively to obtain high-quality representation embedding for classification. Early studies frequently follow BERT by concatenating the `[CLS]` token with the original input x and utilizing this token's vector representation for classification. Another method involves using additional special tokens to enclose the two entities, and then concatenating their embeddings to form the input representation for the relation classification layer (Zhao et al., 2022; Le et al., 2024c).

In this study, we employ the input format suggested by Ma et al. (2024). Specifically, we use a special `[MASK]` token to denote the relation between the head entity (e_h) and tail entity (e_t), and integrate this token with the original sentence x and the two entities. Besides, several learnable tokens are also inserted to avoid relying entirely on hand-crafted tokens. Consequently, the input template is formulated as follows:

$$\begin{aligned}\mathcal{I}(x) = x\,[v_{0:n_0-1}]\,e_h\,[v_{n_0:n_1-1}]\,\texttt{[MASK]}\\ [v_{n_1:n_2-1}]\,e_t\,[v_{n_2:n_3-1}]\,.\end{aligned} \tag{1}$$

where $[v_i]$ represents the *i-th* learnable continuous token, and n_i denotes the length of the token phrases. In our specific implementation, we use a special `[UNUSED]` token as $[v]$. We then forward the templated input $\mathcal{I}(x)$ through a PLM, encoding it into a sequence of continuous vectors. From these, we extract the hidden representation $\boldsymbol{z}_x$ of the input, corresponding to the position of the `[MASK]` token.

$$\boldsymbol{z}_x = f_{\mathcal{M}}(\mathcal{I}(x))[\text{position}(\texttt{[MASK]})], \tag{2}$$

where $f_{\mathcal{M}}(X)$ denotes the forward function of a PLM $\mathcal{M}$ on input X. The latent representation is then used for contrastive learning and predicts the relation associated with the given input x.

3 Methodology

In this section, we present our method aimed at enhancing FCRE by tackling the problem of similar relations. Specifically, we proposed *Clustering Relations via Label Description* (CRLD) to identify groups of similar relations. Accordingly, we propose three loss functions designed to differentiate samples that are semantically similar but belong to distinct relations, using the clustering information.

3.1 Clustering Relations via Label Description

The label description of a relation is a summarization that describes the meaning and provides general information about the relation. It has been demonstrated to be more consistent than the label prototype, which is derived from multiple sample contexts associated with the same label (Li and Lyu, 2024). Therefore, we leverage this information for clustering purposes to recognize similar relations. This framework allows us to identify informative hard negatives for samples using cluster information, thereby enhancing the differentiation of samples in similar classes and potentially improving training convergence (Xiong et al., 2020).

Let $\{(r_i, d_i)\}_{i=1}^N$ denote the set of relations and their corresponding description. For each description d_i, we obtain its embedding $\boldsymbol{d}_i$ by passing it through the same encoder used for input sentences containing entities $f_{\mathcal{M}}(d_i)$, as presented in Section 2.2. However, instead of applying the input template $\mathcal{I}$, we directly use the raw description and

obtain its latent embedding by mean pooling the token representations within the description. Subsequently, we employ the Agglomerative Clustering algorithm (Müllner, 2011) on these embeddings to categorize the relations into K clusters according to their semantic similarity. This clustering method allows us to automatically identify the number of clusters by selecting a distance threshold θ, thereby eliminating the need for manual selection of group sizes for similar relations. Besides, the clustering algorithm is applied iteratively for each batch. Thus, the cluster for each relation is dynamically updated following the encoder's parameters adjustment after each batch, as illustrated in Algorithm 1. As a result, each relation-description pair (r_i, d_i) is assigned to a new cluster $c(r_i) \in \{1, 2, \ldots, K\}$ after each batch.

3.2 Discriminative Loss Functions

In this section, we introduce three innovative loss functions to improve the model's ability to distinguish samples across relations.

Weighted Supervised Contrastive Loss *aims to bring closer positive sample pairs, which share the same relation label, while pushing apart negative pairs that belong to different relation.* To this end, we conduct the Supervised Contrastive Loss (Khosla et al., 2020), enhanced with weighting hard negatives according to the similarity of their label descriptions, thus focusing more on samples from similar relations. Specifically, this loss is computed as follows:

$$\mathcal{L}_{\text{WSC}}(x) = -\sum_{p \in P(x)} \log \frac{f(\boldsymbol{z}_x, \boldsymbol{z}_p)}{\sum_{\bar{x} \in \mathcal{D} \setminus \{x\}} w(x, \bar{x}) \cdot f(\boldsymbol{z}_x, \boldsymbol{z}_{\bar{x}})} \tag{3}$$

where $f(\boldsymbol{z}_x, \boldsymbol{z}_y) = \exp\left(\frac{\gamma(\boldsymbol{z}_x, \boldsymbol{z}_y)}{\tau}\right)$ and

$$w(x, \bar{x}) = \begin{cases} 1 + \alpha \cdot \gamma(\boldsymbol{d}_x, \boldsymbol{d}_{\bar{x}}) & \text{if } c(r_x) = c(r_{\bar{x}}) \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

Here, $\gamma(\cdot, \cdot)$ represents the Cosine Similarity function, τ is the temperature scaling parameter, α is a weighting factor based on the description similarity, and $c(\cdot)$ is the cluster assignment function. $P(x)$ and $\mathcal{D}$ refer to the sets of positive samples for sample x and the entire dataset, respectively.

Cluster-based Mutual Information Loss *aims to bring closer the semantic representation of a sample with its corresponding label description, while pushing apart it from descriptions of different relations.* Specifically, it involves maximizing the mutual information (MI) between the input's hidden representation $\boldsymbol{z}_x$ and its corresponding label description $\boldsymbol{d}_x$, while leveraging cluster information to identify hard negatives. This extends the traditional Mutual Information loss (van den Oord et al., 2019) by introducing a weighting function that considers the similarity between label descriptions within the same cluster.

The mutual information $MI(x)$ between the input embedding $\boldsymbol{z}_x$ and its corresponding label description is lower-bounded by:

$$MI \geq \log B + \text{InfoNCE}(D_B; h), \tag{5}$$

However, we modify the InfoNCE loss by incorporating a weighting function $w(x_i, x_j)$, as presented in equation (4), to assign greater emphasis on hard negatives within the same cluster. The modified InfoNCE is thus defined as follows:

$$\text{InfoNCE}(D_B; h) = \frac{1}{B} \sum_{i=1}^{B} \log \frac{h(\boldsymbol{z}_{x_i}, \boldsymbol{d}_{x_i})}{\sum_{j=1}^{B} w(x_i, x_j) \cdot h(\boldsymbol{z}_{x_i}, \boldsymbol{d}_{x_j})},$$

where $h(\boldsymbol{z}, \boldsymbol{d}) = \exp\left(\frac{\boldsymbol{z}^T W \boldsymbol{d}}{\tau}\right)$, τ is the temperature, $D_B = \{x_i\}_{i=1}^{B}$ is the mini-batch data with size B, and W is a trainable parameter. The final CMI loss function with an input x is defined as:

$$\mathcal{L}_{\text{CMI}}(x) = -\log \frac{h(\boldsymbol{z}_x, \boldsymbol{d}_x)}{Z(x)} \tag{6}$$

where,

$$Z(x) = h(\boldsymbol{z}_x, \boldsymbol{d}_x) + \sum_{\bar{x} \in \mathcal{N}(x)} w(x, \bar{x}) \cdot h(\boldsymbol{z}_x, \boldsymbol{d}_{\bar{x}})$$

$$\mathcal{N}(x) = \{\bar{x} | \bar{x} \in D_B, r_{\bar{x}} \neq r_x\}$$

Double Triplet Loss for Intra- and Inter-Cluster Separation *aims to enforce that samples are close to their own descriptions and far from other cluster centroids, especially those not associated with their labels.* In particular, we employ Double Triplet Loss (DTL) (Schroff et al., 2015) on input, label description, and cluster centroid.

$$\mathcal{L}_{\text{DT}}(x) = \max\left(0, D(\boldsymbol{z}_x, \boldsymbol{d}_x) - D(\boldsymbol{z}_x, \boldsymbol{c}_x^+) + m_1\right) + \max\left(0, D(\boldsymbol{z}_x, \boldsymbol{c}_x^+) - D(\boldsymbol{z}_x, \boldsymbol{c}_x^-) + m_2\right) \tag{7}$$

where,

$$\boldsymbol{c}_x^+ = \boldsymbol{c}_k = \frac{1}{|C_k|} \sum_{d \in C_k} \boldsymbol{d} \quad \text{if } c(r_x) = k$$

$$\boldsymbol{c}_x^- = \boldsymbol{c}_{\bar{k}} \quad \text{such that } \bar{k} = \underset{c(r_x) \neq k}{\text{argmin}}\, D(\boldsymbol{z}_x, \boldsymbol{c}_k)$$

Here, $D(\cdot, \cdot) = 1 - \gamma(\cdot, \cdot)$ is the Cosine Distance, $\boldsymbol{c}_k, C_k$ represent the centroid representation and set containing the description embeddings of relations within cluster k, respectively. $\boldsymbol{c}_x^+$ is the centroid representation of the cluster which containing r_x, while $\boldsymbol{c}_x^-$ denotes the centroid of the nearest cluster to x that does not include the relation r_x. m_1 and m_2 are margin hyperparameters.

Overall Training Objective. The total loss function is formulated by combining equations (3), (6) and (7) as follows:

$$\mathcal{L}(x) = \lambda_1 \mathcal{L}_{\text{WSC}}(x) + \lambda_2 \mathcal{L}_{\text{CMI}}(x) + \lambda_3 \mathcal{L}_{\text{DT}}(x) \tag{8}$$

where λ_1, λ_2 and λ_3 are weighting hyperparameters that balance the contributions of each loss component.

3.3 Large Language Model Embeddings for FCRE

Large Language Model Embeddings (LLMEs) redefine decoder-only LLMs as text encoders, enhancing their embedding and representation learning capabilities (BehnamGhader et al., 2024; Li et al., 2024; Lee et al., 2024). This transformation typically involves two main adjustments: (1) removing the causal mask to allow bidirectional attention, and (2) replacing the next-token prediction task with alternative training objectives, such as contrastive learning or masked token prediction. These modifications enable LLMEs to function similarly to encoder models like BERT while leveraging the extensive architecture and pretraining corpus of the original LLMs, thereby enhancing generalization and comprehension capabilities.

We explore the application of LLMEs in the FCRE scenario by substituting the backbone model $\mathcal{M}$ with these models, as described in Section 2.2. However, since LLMs perform well with instruction prompts and mean-pooling all token embeddings yields the best results in LLM2Vec (BehnamGhader et al., 2024)—an LLME, we formulate an input x with entities e_h, e_t as follows.

$\mathcal{F}_{LLMEs}(x) =$ x. The relation between $[e_h]$ and $[e_t]$ is:

This instruction prompt allows LLMEs to grasp the semantic context to categorize relations for the entities. The latent embedding is subsequently obtained by mean pooling the token representations. Training and inference procedures remain similar across all backbone models.

3.4 Training and Inference Procedures

Algorithm 1 Training procedure at each task $\mathcal{T}^j$

Input:
$\mathcal{M}$: Backbone PLM
$\mathcal{D}^{test}$: Test data
L: The number of training samples allocated to memory for each relation.
Previous variables: $\Phi_{j-1}, \tilde{R}_{j-1}, \tilde{M}_{j-1}, \tilde{S}_{j-1}$
Current variables: $D_j^{train}, D_j^{test}, R_j, S_j$.
Output:
$\Phi_j, \tilde{M}_j, \tilde{S}_j, \tilde{P}_j$.

Initialize Φ_j from Φ_{j-1}
$\tilde{S}_j \leftarrow \tilde{S}_{j-1} \cup S_j$, $\tilde{R}_j \leftarrow \tilde{R}_{j-1} \cup R_j$
for batch in batches($\tilde{M}_{j-1} \cup D_j$) **do**
    $\boldsymbol{d}_i \leftarrow f_{\mathcal{M}_{\Phi_j}}(d_i) \quad \forall d_i \in \tilde{S}_j$
    AGGLOMERATIVECLUSTERING($\{\boldsymbol{d}_i\}$)
    Update Φ_j ▷ Backward using loss $\mathcal{L}$ in (8)
end for
$\tilde{M}_j \leftarrow \tilde{M}_{j-1}$
for each $r \in R_j$ **do** ▷ Update memory buffer
    $\mathcal{B}_r \leftarrow \{(x_i, r_i) | x_i \in D_j^{train}, r_i = r\}_{i=1}^{L}$
    $\tilde{M}_j \leftarrow \tilde{M}_j \cup \mathcal{B}_r$
end for
$\mathcal{D}^{test} \leftarrow \mathcal{D}^{test} \cup D_j^{test}$ ▷ For inference

Training Procedure: Algorithm 1 outlines the end-to-end training process at each task $\mathcal{T}^j$, with Φ_{j-1} denoting the model parameters after training on the previous $j-1$ tasks. In line with memory-based methods, we maintain a memory buffer $\tilde{M}_{j-1}$ that stores a few representative samples from all previous tasks $\mathcal{T}^1, \ldots, \mathcal{T}^{j-1}$, along

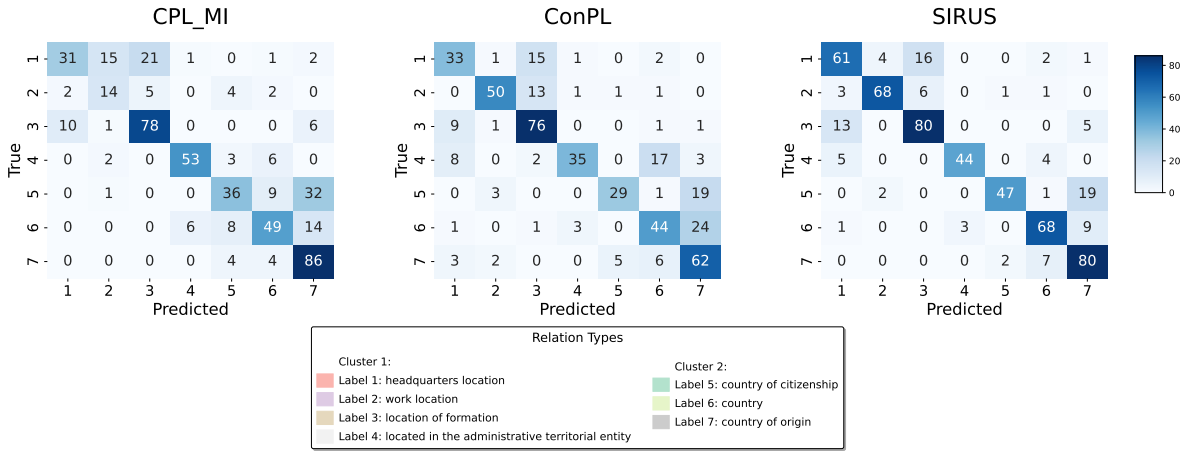

Figure 1: The confusion matrix of CPL_MI, ConPL, and SIRUS predictions on similar relations detected by our CRLD method. The descriptions of relations are presented in Table 8. Note that the figure illustrates only the relations from the test set that appear in these two detected similar clusters, rather than displaying all relations.

with a relation description set $\tilde{S}_{j-1}$ that holds the descriptions of all previously encountered relations.

1. **Initialization** (Lines 1–2): The model parameter for the current task, Φ_j, is initialized with the parameters of Φ_{j-1}. We update the relation $\tilde{R}_j$ and relation description sets $\tilde{E}_j$ with new relations.

2. **Model Update** (Lines 3–7): We train and update the model parameters using a dataset that combines memory data with the data from the current task $\mathcal{T}^j$. Relation description representation and cluster results are updated in each batch (Lines 4–5). Accordingly, the loss function is adjusted to reflect the cluster changes to update the current parameter Φ_j.

3. **Memory Update** (Lines 8–12): We select L representative samples from D_j for each relation $r \in R_j$. These are the L samples whose latent representations are closest to the 1-means centroid of all class samples.

4. **Testset Update** (Line 13): The test set is expanded by incorporating the test data from the current task and will be utilized for evaluation after finishing training across all tasks.

Inference Procedure: Leveraging the discriminative feature distribution learned during training, we adopt the Nearest-Class-Mean classifier, as employed by Ma et al. (2024), for relation prediction in the test phase. However, instead of relying solely on the label prototype, we incorporate both the label description and prototype to extract the relation.

Given a sample x with hidden representation $\boldsymbol{z_x}$, a set of relation prototypes $\{\boldsymbol{p}_r\}_{r=1}^{n}$ and a set of relation descriptions $\{\boldsymbol{d}_r\}_{r=1}^{n}$.

$$\boldsymbol{p}_r = \frac{1}{L}\sum_{i=1}^{L} \boldsymbol{z}_i, \tag{9}$$

The inference process begins by calculating the Cosine similarity between $\boldsymbol{z_x}$ and each prototype $\boldsymbol{p}_r$ and label description $\boldsymbol{d}_r$. The final prediction y^* is then determined by:

$$y^* = \text{argmax}\left(\gamma(\boldsymbol{z}_x, \boldsymbol{p}_r) + \gamma(\boldsymbol{z}_x, \boldsymbol{d}_r)\right) \tag{10}$$

where $\gamma(\cdot,\cdot)$ denotes the cosine similarity function.

4 Experimental Results

4.1 Experiment Setup

We compare our method against 8 state-of-the-art baselines on two widely used benchmarks FewRel (Han et al., 2018) and TACRED (Zhang et al., 2017) in the literature of CRE and FCRE. We conduct experiments using BERT (Devlin et al., 2019) and two LLMEs: LLM2Vec (BehnamGhader et al., 2024) and BGE (Li et al., 2024). Moreover, we employ three variants of LLM2Vec, using LLama2, LLama3, and Mistral as backbones. After completing each task, we evaluate the models on the updated $\mathcal{D}^{test}$ with 6 random seeds and report the mean and standard deviation of the accuracy. More details about datasets, baselines, backbones, and evaluation metrics are presented in Appendix B.

Method	Tasks							
	$\mathcal{T}^1$	$\mathcal{T}^2$	$\mathcal{T}^3$	$\mathcal{T}^4$	$\mathcal{T}^5$	$\mathcal{T}^6$	$\mathcal{T}^7$	$\mathcal{T}^8$
FewRel *(10-way–5-shot)*								
RP-CRE (Cui et al., 2021)	$93.97_{\pm 0.64}$	$76.05_{\pm 2.36}$	$71.36_{\pm 2.83}$	$69.32_{\pm 3.98}$	$64.95_{\pm 3.09}$	$61.99_{\pm 2.09}$	$60.59_{\pm 1.87}$	$59.57_{\pm 1.13}$
CRL (Zhao et al., 2022)	$94.68_{\pm 0.33}$	$80.73_{\pm 2.91}$	$73.82_{\pm 2.77}$	$70.26_{\pm 3.18}$	$66.62_{\pm 2.74}$	$63.28_{\pm 2.49}$	$60.96_{\pm 2.63}$	$59.27_{\pm 1.32}$
CRECL (Hu et al., 2022)	$93.93_{\pm 0.22}$	$82.55_{\pm 6.95}$	$74.13_{\pm 3.59}$	$69.33_{\pm 3.87}$	$66.51_{\pm 4.05}$	$64.60_{\pm 1.92}$	$62.97_{\pm 1.46}$	$59.99_{\pm 0.65}$
ERDA (Qin and Joty, 2022)	$92.43_{\pm 0.32}$	$64.52_{\pm 2.11}$	$50.31_{\pm 3.32}$	$44.92_{\pm 3.77}$	$39.75_{\pm 3.34}$	$36.36_{\pm 3.12}$	$34.34_{\pm 1.83}$	$31.96_{\pm 1.91}$
SCKD (Wang et al., 2023)	$94.77_{\pm 0.35}$	$82.83_{\pm 2.61}$	$76.21_{\pm 1.61}$	$72.19_{\pm 1.33}$	$70.61_{\pm 2.24}$	$67.15_{\pm 1.96}$	$64.86_{\pm 1.35}$	$62.98_{\pm 0.88}$
ConPL§ (Chen et al., 2023)	$\mathbf{95.18_{\pm 0.73}}$	$79.63_{\pm 1.27}$	$74.54_{\pm 1.13}$	$71.27_{\pm 0.85}$	$68.35_{\pm 0.86}$	$63.86_{\pm 2.03}$	$64.74_{\pm 1.39}$	$62.46_{\pm 1.54}$
CPL (Ma et al., 2024)	$\underline{94.87}$	85.14	78.80	75.10	72.57	69.57	66.85	64.50
CPL_MI (Tran et al., 2024)	$94.69_{\pm 0.70}$	$\underline{85.58_{\pm 1.88}}$	$\underline{80.12_{\pm 2.45}}$	$\underline{75.71_{\pm 2.28}}$	$\underline{73.90_{\pm 1.80}}$	$\underline{70.72_{\pm 0.91}}$	$\underline{68.42_{\pm 1.77}}$	$\underline{66.27_{\pm 1.58}}$
SIRUS	$94.74_{\pm 0.27}$	$\mathbf{87.12_{\pm 2.21}}$	$\mathbf{81.06_{\pm 1.52}}$	$\mathbf{77.49_{\pm 2.58}}$	$\mathbf{75.47_{\pm 2.60}}$	$\mathbf{72.48_{\pm 1.75}}$	$\mathbf{70.6_{\pm 1.31}}$	$\mathbf{69.16_{\pm 0.43}}$ ↑ 2.89
TACRED *(5-way-5-shot)*								
RP-CRE (Cui et al., 2021)	$87.32_{\pm 1.76}$	$74.90_{\pm 6.13}$	$67.88_{\pm 4.31}$	$60.02_{\pm 5.37}$	$53.26_{\pm 4.67}$	$50.72_{\pm 7.62}$	$46.21_{\pm 5.29}$	$44.48_{\pm 3.74}$
CRL (Zhao et al., 2022)	$88.32_{\pm 1.26}$	$76.30_{\pm 7.48}$	$69.76_{\pm 5.89}$	$61.93_{\pm 2.55}$	$54.68_{\pm 3.12}$	$50.92_{\pm 4.45}$	$47.00_{\pm 3.78}$	$44.27_{\pm 2.51}$
CRECL (Hu et al., 2022)	$87.09_{\pm 2.50}$	$78.09_{\pm 5.74}$	$61.93_{\pm 4.89}$	$55.60_{\pm 5.78}$	$53.42_{\pm 2.99}$	$51.91_{\pm 2.95}$	$47.55_{\pm 3.38}$	$45.53_{\pm 1.96}$
ERDA (Qin and Joty, 2022)	$81.88_{\pm 1.97}$	$53.68_{\pm 6.31}$	$40.36_{\pm 3.35}$	$36.17_{\pm 3.65}$	$30.14_{\pm 3.96}$	$22.61_{\pm 3.13}$	$22.29_{\pm 1.32}$	$19.42_{\pm 2.31}$
SCKD (Wang et al., 2023)	$\underline{88.42_{\pm 0.83}}$	$79.35_{\pm 4.13}$	$70.61_{\pm 3.16}$	$66.78_{\pm 4.29}$	$60.47_{\pm 3.05}$	$58.05_{\pm 3.84}$	$54.41_{\pm 3.47}$	$52.11_{\pm 3.15}$
ConPL§ (Chen et al., 2023)	$\mathbf{88.77_{\pm 0.84}}$	$69.64_{\pm 1.93}$	$57.50_{\pm 2.48}$	$52.15_{\pm 1.59}$	$58.19_{\pm 2.31}$	$55.01_{\pm 3.12}$	$52.88_{\pm 3.66}$	$50.97_{\pm 3.41}$
CPL (Ma et al., 2024)	86.27	81.55	73.52	68.96	63.96	62.66	59.96	57.39
CPL_MI (Tran et al., 2024)	$85.67_{\pm 0.80}$	$\underline{82.54_{\pm 2.98}}$	$\underline{75.12_{\pm 3.67}}$	$\underline{70.65_{\pm 2.75}}$	$\underline{66.79_{\pm 2.18}}$	$\underline{65.17_{\pm 2.48}}$	$\underline{61.25_{\pm 1.52}}$	$\underline{59.48_{\pm 3.53}}$
SIRUS	$87.41_{\pm 0.41}$	$\mathbf{84.28_{\pm 7.38}}$	$\mathbf{76.38_{\pm 3.99}}$	$\mathbf{73.86_{\pm 4.16}}$	$\mathbf{68.06_{\pm 5.57}}$	$\mathbf{66.64_{\pm 5.76}}$	$\mathbf{62.74_{\pm 3.92}}$	$\mathbf{60.68_{\pm 3.53}}$ ↑ 1.2

Table 1: Accuracy (%) of methods using BERT backbone after training for each task. The best results are in **bold**, while the second highest scores are underlined. All the baseline results are obtained from (Wang et al., 2023) and (Tran et al., 2024). § ConPL results that are reproduced with the same settings as other models (Appendix B.2).

.

4.2 Evaluation Results of SIRUS Framework

In this section, we analyze the results of Clustering Relations via Label Description (CRLD) and our proposed SIRUS, which integrates CRLD with three loss functions in FCRE.

Effectiveness of CRLD: Figure 1 presents the relations within two clusters that are close together generated by CRLD, showing that they share a common topic related to location and country. Through the visualization of the confusion matrix in these relations derived by 3 methods, we observe that these relations are frequently confused. For instance, models often misclassify samples to the relations "*country of origin*" and "*country of citizenship*" or "*headquarters location*" and "*location of formation*". This observation indicates that CRLD can effectively identify similar relations.

In addition, in comparison to CPL_MI, ConPL– a method that also tackles the challenge of similar relations, offers a clearer differentiation among these relations. For example, between two relations "*country*" and "*country of origin*", CPL_MI depicts a higher number of misclassified samples than ConPL (32 compared to 19). Meanwhile, SIRUS shows clearer results in reducing confusion between these classes, with darker blocks along the diagonal (indicating more correct classifications) and lighter blocks outside the diagonal (reflecting fewer misclassifications). This not only demonstrates the effectiveness of CRLD in accurately identifying similar relations, which aids the model in focusing on learning to differentiate them but also reveals the overall efficacy of SIRUS for FCRE.

Performance of SIRUS against the Baselines: Table 1 presents the performance comparison between SIRUS and 8 state-of-the-art methods. Overall, our approach consistently outperforms the performance of existing methods across all tasks on both benchmarks. Specifically, on the FewRel dataset, SIRUS surpasses the baselines by a significant margin, achieving over 1% improvement compared to the second-best method, CPL_MI, across all tasks. As a result, after the final task, SIRUS achieves 69.16% accuracy on the test set encompassing all relations, demonstrating an improvement of nearly 3% over CPL_MI. A similar trend is observed in the TACRED dataset, where our method outperforms CPL_MI by more than 1%. Besides, additional results demonstrating the effectiveness of each proposed loss function are provided in Appendix C.1. These results demonstrate the effectiveness of SIRUS and underscore the importance of addressing the issue of similar relations in FCRE.

Backbone	Method	LLME	Tasks							
			$\mathcal{T}^1$	$\mathcal{T}^2$	$\mathcal{T}^3$	$\mathcal{T}^4$	$\mathcal{T}^5$	$\mathcal{T}^6$	$\mathcal{T}^7$	$\mathcal{T}^8$
FewRel *(10-way–5-shot)*										
LLama2	CPL	✗	95.73$_{\pm 0.92}$	85.87$_{\pm 1.46}$	80.57$_{\pm 1.74}$	78.60$_{\pm 3.31}$	77.30$_{\pm 2.41}$	73.95$_{\pm 1.54}$	71.35$_{\pm 3.75}$	69.87$_{\pm 2.32}$
	CPL_MI	✗	95.63$_{\pm 1.08}$	87.14$_{\pm 1.94}$	83.25$_{\pm 2.14}$	80.59$_{\pm 2.37}$	79.20$_{\pm 1.36}$	76.41$_{\pm 2.13}$	74.62$_{\pm 1.73}$	72.08$_{\pm 3.18}$
	CPL†	✓	**96.78$_{\pm 0.37}$**	**88.91$_{\pm 3.38}$**	**84.90$_{\pm 2.93}$**	**81.99$_{\pm 2.21}$**	79.20$_{\pm 3.28}$	77.60$_{\pm 2.4}$	75.57$_{\pm 3.2}$	74.12$_{\pm 1.23}$
	LLM2Vec$^{\dagger}_{\text{w/o-mem}}$	✓	95.05$_{\pm 0.21}$	84.88$_{\pm 2.79}$	78.15$_{\pm 2.54}$	72.4$_{\pm 2.26}$	72.36$_{\pm 2.95}$	69.21$_{\pm 2.81}$	64.89$_{\pm 1.68}$	63.38$_{\pm 0.65}$
	SIRUS †	✓	95.58$_{\pm 0.24}$	87.93$_{\pm 2.54}$	83.22$_{\pm 1.51}$	81.76$_{\pm 1.01}$	**81.23$_{\pm 2.25}$**	**79.07$_{\pm 1.69}$**	**76.86$_{\pm 1.63}$**	**75.98$_{\pm 0.5}$**
Mistral	CPL	✗	96.57$_{\pm 0.40}$	86.80$_{\pm 2.53}$	83.31$_{\pm 1.94}$	79.45$_{\pm 2.53}$	77.17$_{\pm 2.2}$	74.24$_{\pm 1.96}$	73.59$_{\pm 2.00}$	71.89$_{\pm 1.97}$
	CPL_MI	✗	96.55$_{\pm 0.43}$	**90.77$_{\pm 2.11}$**	84.81$_{\pm 1.09}$	83.08$_{\pm 1.50}$	78.92$_{\pm 1.35}$	77.27$_{\pm 2.06}$	77.05$_{\pm 2.30}$	75.02$_{\pm 1.67}$
	CPL†	✓	**96.6$_{\pm 0.22}$**	88.75$_{\pm 2.63}$	84.39$_{\pm 2.65}$	82.46$_{\pm 2.08}$	80.38$_{\pm 1.93}$	78.06$_{\pm 1.18}$	75.41$_{\pm 1.9}$	74.00$_{\pm 1.32}$
	LLM2Vec$_{\dagger \text{w/o-mem}}$	✓	96.37$_{\pm 0.16}$	86.53$_{\pm 3.87}$	80.50$_{\pm 2.38}$	76.00$_{\pm 1.85}$	72.83$_{\pm 5.08}$	68.50$_{\pm 3.93}$	67.38$_{\pm 2.75}$	65.65$_{\pm 1.43}$
	SIRUS †	✓	96.13$_{\pm 0.31}$	**89.74$_{\pm 2.69}$**	**86.10$_{\pm 2.41}$**	**84.25$_{\pm 2.25}$**	**81.96$_{\pm 2.81}$**	**79.79$_{\pm 2.56}$**	**77.75$_{\pm 2.09}$**	**76.96$_{\pm 1.15}$**
	BGE$^{\ddagger}_{\text{w/o-mem}}$	✓	96.38$_{\pm 0.20}$	86.88$_{\pm 2.48}$	79.58$_{\pm 2.40}$	76.50$_{\pm 1.67}$	73.40$_{\pm 4.04}$	72.80$_{\pm 2.34}$	69.31$_{\pm 2.05}$	67.51$_{\pm 1.89}$
	CPL‡	✓	96.52$_{\pm 0.26}$	89.88$_{\pm 3.33}$	84.3$_{\pm 1.97}$	81.5$_{\pm 2.94}$	79.05$_{\pm 4.02}$	77.27$_{\pm 3.49}$	75.6$_{\pm 2.99}$	73.25$_{\pm 2.48}$
	SIRUS ‡	✓	**96.90$_{\pm 0.34}$**	**91.14$_{\pm 1.83}$**	**87.94$_{\pm 1.46}$**	**86.39$_{\pm 2.11}$**	**84.62$_{\pm 2.22}$**	**82.82$_{\pm 1.96}$**	**80.9$_{\pm 0.69}$**	**79.38$_{\pm 0.48}$**
LLama3	LLM2Vec$^{\dagger}_{\text{w/o-mem}}$	✓	97.25$_{\pm 0.31}$	86.67$_{\pm 3.13}$	80.14$_{\pm 1.27}$	76.12$_{\pm 2.39}$	72.71$_{\pm 3.45}$	68.30$_{\pm 3.71}$	65.15$_{\pm 4.45}$	63.42$_{\pm 4.24}$
	CPL†	✓	**97.37$_{\pm 0.15}$**	87.96$_{\pm 2.66}$	83.02$_{\pm 1.34}$	79.78$_{\pm 2.78}$	78.09$_{\pm 3.09}$	75.95$_{\pm 1.87}$	74.65$_{\pm 1.60}$	73.19$_{\pm 1.11}$
	SIRUS †	✓	96.80$_{\pm 0.18}$	**91.04$_{\pm 2.43}$**	**87.36$_{\pm 1.49}$**	**85.25$_{\pm 1.48}$**	**84.28$_{\pm 2.69}$**	**82.46$_{\pm 1.67}$**	**81.03$_{\pm 1.42}$**	**78.82$_{\pm 0.98}$**
TACRED *(5-way-5-shot)*										
LLama2	CPL	✗	86.76$_{\pm 1.58}$	75.94$_{\pm 4.76}$	70.65$_{\pm 2.57}$	68.64$_{\pm 3.03}$	67.44$_{\pm 2.95}$	65.12$_{\pm 3.85}$	60.27$_{\pm 3.79}$	58.03$_{\pm 1.98}$
	CPL_MI	✗	85.55$_{\pm 0.74}$	77.91$_{\pm 2.80}$	76.49$_{\pm 2.79}$	74.99$_{\pm 2.69}$	69.15$_{\pm 3.65}$	68.19$_{\pm 2.29}$	64.19$_{\pm 3.01}$	62.04$_{\pm 1.10}$
	CPL†	✓	87.37$_{\pm 1.85}$	82.74$_{\pm 9.54}$	77.49$_{\pm 7.52}$	**77.29$_{\pm 4.49}$**	72.75$_{\pm 6.28}$	73.37$_{\pm 4.57}$	70.08$_{\pm 6.01}$	68.35$_{\pm 5.02}$
	LLM2Vec$^{\dagger}_{\text{w/o-mem}}$	✓	88.56$_{\pm 0.66}$	82.34$_{\pm 8.49}$	71.12$_{\pm 4.69}$	68.58$_{\pm 3.06}$	63.82$_{\pm 4.7}$	60.79$_{\pm 3.63}$	55.72$_{\pm 3.77}$	52.99$_{\pm 2.42}$
	SIRUS †	✓	**89.62$_{\pm 0.31}$**	**87.07$_{\pm 7.02}$**	**78.98$_{\pm 4.58}$**	76.04$_{\pm 3.28}$	**74.64$_{\pm 3.15}$**	**74.14$_{\pm 2.39}$**	**70.96$_{\pm 1.77}$**	**70.88$_{\pm 0.59}$**
Mistral	CPL	✗	86.67$_{\pm 0.81}$	80.98$_{\pm 5.42}$	77.16$_{\pm 4.96}$	73.24$_{\pm 3.63}$	70.05$_{\pm 2.5}$	67.70$_{\pm 3.95}$	67.04$_{\pm 3.12}$	64.11$_{\pm 3.68}$
	CPL_MI	✗	86.32$_{\pm 1.25}$	81.00$_{\pm 3.20}$	77.71$_{\pm 2.31}$	75.48$_{\pm 2.59}$	71.92$_{\pm 3.09}$	71.02$_{\pm 2.84}$	67.69$_{\pm 3.58}$	65.48$_{\pm 1.97}$
	CPL†	✓	88.56$_{\pm 0.58}$	83.57$_{\pm 5.25}$	75.54$_{\pm 6.82}$	74.82$_{\pm 5.28}$	72.55$_{\pm 4.99}$	71.13$_{\pm 6.44}$	69.05$_{\pm 5.94}$	67.36$_{\pm 4.67}$
	LLM2Vec$^{\dagger}_{\text{w/o-mem}}$	✓	**89.26$_{\pm 0.37}$**	**84.30$_{\pm 6.65}$**	77.78$_{\pm 2.89}$	72.21$_{\pm 3.78}$	67.66$_{\pm 4.02}$	66.46$_{\pm 3.00}$	63.13$_{\pm 4.58}$	59.68$_{\pm 1.87}$
	SIRUS †	✓	88.24$_{\pm 0.23}$	83.29$_{\pm 5.02}$	**79.12$_{\pm 3.98}$**	**76.92$_{\pm 3.74}$**	**75.26$_{\pm 3.24}$**	**75.31$_{\pm 1.4}$**	**73.64$_{\pm 4.97}$**	**73.06$_{\pm 3.23}$**
	BGE$^{\ddagger}_{\text{w/o-mem}}$	✓	**89.30$_{\pm 0.35}$**	83.09$_{\pm 5.81}$	74.28$_{\pm 3.59}$	70.89$_{\pm 4.88}$	65.44$_{\pm 5.90}$	64.75$_{\pm 3.71}$	61.44$_{\pm 6.42}$	58.85$_{\pm 2.58}$
	CPL‡	✓	88.43$_{\pm 0.86}$	85.51$_{\pm 5.27}$	77.76$_{\pm 5.19}$	**75.79$_{\pm 4.57}$**	**74.23$_{\pm 2.93}$**	71.97$_{\pm 4.47}$	70.68$_{\pm 5.11}$	67.52$_{\pm 5.37}$
	SIRUS ‡	✓	88.33$_{\pm 0.14}$	**85.72$_{\pm 3.46}$**	**78.80$_{\pm 2.80}$**	75.23$_{\pm 3.62}$	73.62$_{\pm 2.18}$	**72.34$_{\pm 1.64}$**	**70.18$_{\pm 3.57}$**	**70.07$_{\pm 2.74}$**
LLama3	LLM2Vec$^{\dagger}_{\text{w/o-mem}}$	✓	88.42$_{\pm 0.41}$	82.51$_{\pm 4.26}$	75.61$_{\pm 2.48}$	72.31$_{\pm 2.94}$	68.22$_{\pm 4.40}$	63.35$_{\pm 4.22}$	59.48$_{\pm 4.57}$	56.94$_{\pm 3.60}$
	CPL†	✓	**88.75$_{\pm 0.59}$**	81.18$_{\pm 9.26}$	76.14$_{\pm 4.36}$	76.16$_{\pm 5.6}$	72.14$_{\pm 5.85}$	71.35$_{\pm 5.41}$	69.99$_{\pm 5.21}$	69.70$_{\pm 5.36}$
	SIRUS †	✓	87.76$_{\pm 0.61}$	**85.85$_{\pm 3.97}$**	**82.19$_{\pm 4.19}$**	**77.61$_{\pm 2.67}$**	**74.86$_{\pm 3.41}$**	**75.67$_{\pm 3.03}$**	**74.42$_{\pm 4.02}$**	**73.97$_{\pm 3.71}$**

Table 2: Accuracy (%) of methods using LLM and LLME-based backbones after training for each task. *w/o-mem* denotes that the memory buffer is excluded during training across tasks. *LLME* column indicates the use of LLMEs (✓) or original LLMs with causal mask (✗). † denotes LLM2Vec variant, while ‡ represents BGE variant. The baseline results of original LLMs with the causal mask are obtained from Tran et al. (2024).

4.3 Evaluation Results of LLMEs in FCRE

In this section, we analyze the results of LLMEs, concentrating on the catastrophic forgetting in these models and comparing their performance to the use of BERT and LLMs with causal language modeling. Additionally, we also aim to assess the effectiveness and adaptability of our method, SIRUS, on these large-scale models.

Catastrophic Forgetting in LLMEs: To investigate the issue of Catastrophic Forgetting in LLMEs, we employ LLM2Vec, BGE-variants as backbones, training them with contrastive loss (Khosla et al., 2020) across sequential tasks while excluding the memory buffer for storing previous data. Table 2 indicates that these models still suffer from catastrophic forgetting. Specifically, their performance remarkably decrease in later tasks. This phenomenon is also observed when utilizing causal LLMs in FCRE (Tran et al., 2024).

Comparison between LLMEs and LLMs: Tables 1 and 2 reveal a significant improvement of up to 10% on both benchmarks in LLMEs compared to BERT when using SIRUS and CPL, highlighting the huge potential of LLMEs in FCRE. Consider utilizing original LLMs with decoder-only architecture, CPL performs worse than CPL_MI by a large margin using both LLama2 and Mistral on the two benchmarks. However, after integrating LLMEs into CPL, it surpasses CPL_MI with the original LLMs by over 2% on FewRel and 6% on TACRED. This result demonstrates the effectiveness of LLMEs' representation and embedding capabilities, extending beyond the generation-focused nature of LLMs in the FCRE scenario.

Our proposed method, SIRUS, consistently outperforms CPL across all cases when integrated with

LLMEs. This showcases the versatility and adaptability of our approach across a diverse range of architectures. Among all backbones, the LLama3 variant of LLM2Vec integrated with SIRUS exhibits the highest performance on FewRel, while BGE with Mistral shows superior results for TACRED.

5 Conclusion

In conclusion, our novel approach to Few-shot Continual Relation Extraction (FCRE) effectively addresses the challenge of similar relations, which often leads to catastrophic forgetting. By leveraging relation descriptions and dynamic clustering, we enhance the distinction between relations through innovative loss functions. Our experimental results indicate that our approach achieves superior performance, surpassing state-of-the-art methods. Moreover, our comprehensive investigation of Large Language Model Embeddings (LLMEs) demonstrates superior performance over both BERT and decoder-only LLMs in all cases, emphasizing their potential to advance FCRE systems. This work paves the way for more robust and accurate relation extraction systems, contributing to the broader field of information extraction.

6 Limitations

Currently, the approach and analyses conducted in this study are limited to high-level relation extraction tasks, where the entities are predetermined. Therefore, to achieve more practical and advancing FCRE systems, it is essential to investigate end-to-end relation extraction challenges in future research, integrating both entity recognition and the extraction of relations among the identified entities. This scenario presents greater challenges as it necessitates addressing both overfitting and catastrophic forgetting across two consecutive tasks.

Our approach primarily targets the challenge of similar relations, particularly leading to catastrophic forgetting; however, it has not yet considered the issue of overfitting, which arises from the constraints of limited data. Despite this limitation, SIRUS demonstrates superior performance compared to techniques that involve augmenting data for previously learned tasks (Qin and Joty, 2022; Ma et al., 2024; Tran et al., 2024). This suggests that while our current method effectively addresses the problem of similar relations, there remains room for improvement. We believe that incorporating data augmentation could further enhance the performance of our method. Therefore, we plan to investigate this approach in future research.

Acknowledgements

This research is funded by Hanoi University of Science and Technology (HUST) under project number T2024-TN-003.

References

Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. 2019. Uncertainty-based continual learning with adaptive regularization. *Advances in neural information processing systems*, 32.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Philipp Borchert, Jochen De Weerdt, and Marie-Francine Moens. 2024. Efficient information extraction in few-shot relation classification through contrastive representation learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 638–646, Mexico City, Mexico. Association for Computational Linguistics.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. Rebel: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381.

Xiudi Chen, Hui Wu, and Xiaodong Shi. 2023. Consistent prototype learning for few-shot continual relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7409–7422, Toronto, Canada. Association for Computational Linguistics.

Li Cui, Deqing Yang, Jiaxin Yu, Chengwei Hu, Jiayang Cheng, Jingjie Yi, and Yanghua Xiao. 2021. Refining sample embeddings with relation prototypes to enhance continual relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 232–243.

Viet Dao, Van-Cuong Pham, Quyen Tran, Thanh-Thien Le, Linh Ngo, and Thien Nguyen. 2024. Lifelong event detection via optimal transport. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12610–12621.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nam Le Hai, Trang Nguyen, Linh Ngo Van, Thien Huu Nguyen, and Khoat Than. 2024. Continual variational dropout: a view of auxiliary local variables in continual learning. *Machine Learning*, 113(1):281–323.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 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*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Douglas M Hawkins. 2004. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *arXiv preprint arXiv:2103.06268*.

Chengwei Hu, Deqing Yang, Haoliang Jin, Zhen Chen, and Yanghua Xiao. 2022. Improving continual relation extraction through prototypical contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1885–1895, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Bin Ji, Jie Yu, Shasha Li, Jun Ma, Qingbo Wu, Yusong Tan, and Huijun Liu. 2020. Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 88–99, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Sangwon Jung, Hongjoon Ahn, Sungmin Cha, and Taesup Moon. 2020. Continual learning with node-importance based adaptive group sparse regularization. *Advances in neural information processing systems*, 33:3647–3658.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Ruben Kruiper, Julian Vincent, Jessica Chen-Burger, Marc Desmulliez, and Ioannis Konstas. 2020. In layman's terms: Semi-open relation extraction from scientific texts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1489–1500.

Minh Le, Tien Ngoc Luu, An Nguyen The, Thanh-Thien Le, Trang Nguyen, Tung Thanh Nguyen, Linh Ngo Van, and Thien Huu Nguyen. 2025. Adaptive prompting for continual relation extraction: A within-task variance perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Minh Le, An Nguyen, Huy Nguyen, Trang Nguyen, Trang Pham, Linh Van Ngo, and Nhat Ho. 2024a. Mixture of experts meets prompt-based continual learning. In *Advances in Neural Information Processing Systems*.

Thanh-Thien Le, Viet Dao, Linh Nguyen, Thi-Nhung Nguyen, Linh Ngo, and Thien Nguyen. 2024b. Sharpseq: Empowering continual event detection through sharpness-aware sequential-task learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3632–3644.

Thanh-Thien Le, Manh Nguyen, Tung Thanh Nguyen, Linh Ngo Van, and Thien Huu Nguyen. 2024c. Continual relation extraction via sequential multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18444–18452.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners. *arXiv preprint arXiv:2409.15700*.

Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. 2019. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International conference on machine learning*, pages 3925–3934. PMLR.

Zhiming Li and Yuchen Lyu. 2024. GRADUAL: Granularity-aware dual prototype learning for better few-shot relation extraction. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13566–13577, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Da Luo, Yanglei Gan, Rui Hou, Run Lin, Qiao Liu, Yuxiang Cai, and Wannian Gao. 2024. Synergistic anchored contrastive pre-training for few-shot relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18742–18750.

Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N Arighi, and Zhiyong Lu. 2022. Biored: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, 23(5):bbac282.

Shengkun Ma, Jiale Han, Yi Liang, and Bo Cheng. 2024. Making pre-trained language models better continual few-shot relation extractors. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10970–10983, Torino, Italia. ELRA and ICCL.

Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms.

Huy Nguyen, Chien Nguyen, Linh Ngo, Anh Luu, and Thien Nguyen. 2023. A spectral viewpoint on continual relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9621–9629.

Xuan Thanh Nguyen, Duc Le Anh, Tran Quyen, Le Thanh-Thien, Linh Ngo Van, and Thien Huu Nguyen. 2025. Few-shot, no problem: Descriptive continual relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hoang Phan, Anh Phan Tuan, Son Nguyen, Ngo Van Linh, and Khoat Than. 2022. Reducing catastrophic forgetting in neural networks via gaussian mixture approximation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 106–117. Springer.

Chengwei Qin and Shafiq Joty. 2022. Continual few-shot relation learning via embedding space regularization and data augmentation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2776–2789, Dublin, Ireland. Association for Computational Linguistics.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in neural information processing systems*, 32.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero and few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1199–1212, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2020. Recurrent interaction network for jointly extracting entities and classifying relations. *arXiv preprint arXiv:2005.00162*.

Wei Tang, Benfeng Xu, Yuyue Zhao, Zhendong Mao, Yifeng Liu, Yong Liao, and Haiyong Xie. 2022. Unirel: Unified representation and interaction for joint relational triple extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7087–7099.

Sebastian Thrun and Tom M Mitchell. 1995. Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46.

Quyen Tran, Nguyen Xuan Thanh, Nguyen Hoang Anh, Nam Le Hai, Trung Le, Linh Van Ngo, and Thien Huu Nguyen. 2024. Preserving generalization of language models in few-shot continual relation extraction.

Linh Ngo Van, Nam Le Hai, Hoang Pham, and Khoat Than. 2022. Auxiliary local variables for improving regularization/prior approach in continual learning. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 16–28. Springer.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Peiyi Wang, Yifan Song, Tianyu Liu, Binghuai Lin, Yunbo Cao, Sujian Li, and Zhifang Sui. 2022. Learning robust representations for continual relation extraction via adversarial class augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6264–6278.

Xinyi Wang, Zitao Wang, and Wei Hu. 2023. Serial contrastive knowledge distillation for continual few-shot relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12693–12706, Toronto, Canada. Association for Computational Linguistics.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Kang Zhao, Hua Xu, Jiangong Yang, and Kai Gao. 2022. Consistent representation learning for continual relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3402–3411, Dublin, Ireland. Association for Computational Linguistics.

Wenzheng Zhao, Yuanning Cui, and Wei Hu. 2023. Improving continual relation extraction by distinguishing analogous semantics. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1162–1175, Toronto, Canada. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

Appendix

A Related work

Continual Learning (CL) aims to progressively learn new knowledge from a sequence of tasks while preventing the problem of forgetting learned knowledge, known as catastrophic forgetting (Thrun and Mitchell, 1995). Several approaches have been explored and can be classified into three main categories: regularization/prior-based methods (Kirkpatrick et al., 2017; Ahn et al., 2019; Jung et al., 2020), architecture-based methods (Li et al., 2019), and memory-based methods (Shin et al., 2017; Rolnick et al., 2019). Memory-based methods, which store a limited number of representative samples from the current task and replay them after subsequent tasks to reinforce prior knowledge, have become widely adopted in NLP tasks, especially in relation extraction (Cui et al., 2021; Zhao et al., 2022; Hu et al., 2022).

Few-shot Continual Relation Extraction (FCRE) aligns with the scope of continual relation extraction research, but faces the additional challenge of limited sample availability for newly emerging relations. Therefore, it poses challenges related to both overfitting and catastrophic forgetting. The concept was first introduced by Qin and Joty (2022), and they introduced a data augmentation framework to address the challenges of data scarcity and catastrophic forgetting. Subsequently, several studies on FCRE have been introduced (Wang et al., 2023; Chen et al., 2023; Ma et al., 2024; Luo et al., 2024; Tran et al., 2024), most of which primarily rely on the memory-based approach. In particular, Wang et al. (2023) employs serial knowledge distillation and contrastive learning, while Chen et al. (2023) introduces a framework comprising three key modules: a prototype-based classification module, a memory-enhanced module, and a consistent learning module. Meanwhile, Luo et al. (2024) improves the contrastive loss component with a multi-view perspective, serving label and instance as distinct anchors, thereby enhancing representation learning for few-shot scenarios. Recently, Tran et al. (2024) investigated the potential of LLMs in FCRE, employing mutual information maximization on the language model head to retain prior knowledge.

B Experimental Details

B.1 Datasets

We conduct our experiments on two benchmark datasets:

- **FewRel** (Han et al., 2018) consists of 100 relations and 70,000 examples. Following the setup in Qin and Joty (2022), we use 80 relations, divided into 8 tasks, each containing 10 relations *(10-way)*. The first task $\mathcal{T}^1$ contains 100 examples per relation, while the remaining tasks are few-shot tasks performed under a *5-shot* setting.
- **TACRED** (Zhang et al., 2017) features 42 relations with 106,264 examples sourced from Newswire and Web documents. In line with the methodology from (Qin and Joty, 2022), we remove instances labeled as "no_relation" and distribute the remaining 41 relations across 8 tasks. Task $\mathcal{T}^1$ includes 6 relations, each with 100 examples, and the following tasks are *5-way 5-shot* tasks, each involving 5 relations.

B.2 Baselines

In this section, we provide a brief overview of several state-of-the-art methods in Few-Shot Continual Relation Extraction (FCRE) that serve as benchmark baselines in our evaluations, including:

- **SCKD** (Wang et al., 2023) implements a structured approach to knowledge distillation, focusing on retaining knowledge from earlier tasks. Additionally, this method leverages contrastive learning with pseudo-samples to improve the differentiation between representations of various relations.
- **CPL** (Ma et al., 2024) introduces a Contrastive Prompt Learning framework, which designs prompts to generalize across relation categories and applies margin-based contrastive learning to manage challenging samples. This helps reduce both catastrophic forgetting and overfitting. The method also incorporates a memory augmentation strategy by generating diverse samples using ChatGPT, which alleviates overfitting in low-resource Few-Shot Continual Relation Extraction scenarios.

- **RP-CRE** (Cui et al., 2021): This method addresses Continual Relation Extraction (CRE) by utilizing stored samples to reduce the forgetting of previously learned relations. It applies K-means clustering to generate prototypes that represent each relation based on the stored data. These prototypes are then used to adjust the embeddings of new samples, allowing the model to retain knowledge of past relations while learning new ones. This approach improves memory efficiency compared to earlier CRE models, leading to better performance.

- **CRL** (Zhao et al., 2022): This approach tackles catastrophic forgetting by implementing a consistent representation learning strategy. It focuses on maintaining stable relation embeddings through contrastive learning and knowledge distillation during the replay of stored samples. The method applies supervised contrastive learning on a memory bank dedicated to each new task, followed by contrastive replay of memory samples and knowledge distillation to preserve knowledge of previous relations. This consistent representation learning effectively mitigates forgetting.

- **CRECL** (Hu et al., 2022): This method enhances traditional few-shot learning by introducing additional constraints on the training data. It achieves this by incorporating information from support instances to enrich instance representations. Additionally, it promotes open-source task enrichment to enable cross-domain knowledge aggregation and introduces the TinyRel-CM dataset, specifically designed for few-shot relation classification with limited training data. Experimental results demonstrate its effectiveness in improving performance in low-data scenarios.

- **ERDA** (Qin and Joty, 2022): This work introduces Continual Few-Shot Relation Learning (CFRL) as a new challenge, highlighting the limitations of existing methods that require extensive labeled data for new tasks. CFRL aims to learn new relations with minimal data while avoiding catastrophic forgetting. To address this, ERDA proposes a technique based on embedding space regularization and data augmentation. This approach enforces constraints on relational embeddings and supplements relevant data through self-supervision. Comprehensive experiments demonstrate that ERDA significantly outperforms previous state-of-the-art methods in CFRL settings.

- **ConPL** (Chen et al., 2023) presents a method with three key components: a prototype-based classification module, a memory-enhanced module, and a consistent learning module aimed at preserving distribution consistency and minimizing forgetting. Additionally, ConPL utilizes prompt learning to improve representation learning and incorporates focal loss to reduce confusion between closely related classes.

- **CPL+MI** (Tran et al., 2024) introduces an innovative approach to improve FCRE models by effectively utilizing the language model (LM) heads. By maximizing the mutual information between these heads and the primary classifiers, the method better preserves prior knowledge from pre-trained backbones while also enhancing representation learning.

It is important to note that we reproduce the results of ConPL (Chen et al., 2023) using the same settings as SCKD and CPL. This adjustment is made because the evaluation strategy in the original paper is not feasible for continual learning scenarios.

B.3 Pre-trained language models

- For BERT-based models: We use BERT-base-uncased checkpoint[1] on Hugging Face.

- For LLM2Vec-based models: We use three checkpoints on Huggingface:

  – Meta-Llama-3-8B-Instruct-mntp-supervised [2],

  – LLM2Vec-Mistral-7B-Instruct-v2-mntp-unsup-simcse [3]

[1] https://huggingface.co/bert-base-uncased

[2] https://huggingface.co/McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp-supervised

[3] https://huggingface.co/McGill-NLP/LLM2Vec-Mistral-7B-Instruct-v2-mntp-unsup-simcse

– LLM2Vec-Llama-2-7b-chat-hf-mntp-supervised [4] checkpoint on Hugging Face.

- For BGE models: We use the bge-en-icl [5] checkpoint on Hugging Face.

B.4 Evaluation and Training Configurations

For each reported result, we conduct 6 independent runs with different random seeds and report the mean and the corresponding standard deviation.

Evaluation Metric: We use final average accuracy to evaluate methods in our experiments. The average accuracy after training task T_j is calculated as follows:

$$ACC_j = \frac{1}{j}\sum_{i=1}^{j} ACC_{j,i}$$

where $ACC_{j,i}$ is the accuracy on the test set of task T_i after training the model on task T_j.

Training Configuration: Our BERT-based experiments were conducted on an NVIDIA RTX 3090 GPU with 24GB of memory. For experiments with the LLM2Vec and BGE backbone, we utilized an NVIDIA A100 GPU with 80GB of VRAM. The operating system used across all experiments was Ubuntu Server 18.04.3 LTS.

Details of hyperparameter search:

- Learning rate: {$\mathbf{1 \times 10^{-5}}$, 2×10^{-5}, $\mathbf{1 \times 10^{-4}}$}
- α: { 0.1, 0.15, **0.2**, **0.25**}
- λ_1: {0.5, **1.0**, 1.5, 2.0, 2.5}
- λ_2: {0.5, **1.0**, 1.5, **2.0**, 2.5}
- λ_3: {**0.25**, **0.5**, 0.75, 1.0}
- τ $\mathcal{L}_{\text{CMI}}$: {**0.01**, 0.02, 0.03, 0.04, **0.05**}
- m_1: { **1.0**, 2.0}
- m_2: { **1.0**, 2.0}
- θ: { **0.1**, 0.2, **0.3**, 0.4, **0.5**, 0.6, 0.7, 0.8}

Lora config target modules: "q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj".

Additionally, Tables 3 and 4 provide the optimal values of hyperparameters for each model backbone.

C Additional experimental results

C.1 Ablation study

Effectiveness of Each Loss Component: Figure 1 and Table 1 have demonstrated the effectiveness of our proposed loss functions in enhancing discriminative representations for similar relations, thereby mitigating catastrophic forgetting and improving overall performance. We further investigate the impact of each loss component on performance through an ablation study by removing each component from the total loss function. Table 5 demonstrates that the absence of any loss component leads to a drop in performance. Specifically, $\mathcal{L}_{CMI}$ (6) exhibits a significant impact, underscoring the importance of leveraging label descriptions to effectively distinguish similar relations. Besides, the results present

[4] https://huggingface.co/McGill-NLP/LLM2Vec-Llama-2-7b-chat-hf-mntp-supervised
[5] https://huggingface.co/BAAI/bge-en-icl

Table 3: Hyperparameters setting for the BERT-backbone.

Hyperparameter	Value
Epochs	10
Learning rate	1×10^{-5}
α	0.25
τ $\mathcal{L}_{\text{CMI}}$ (FewRel)	0.01
τ $\mathcal{L}_{\text{CMI}}$ (TACRED)	0.05
τ $\mathcal{L}_{\text{WSC}}$	0.1
θ (TACRED)	0.3
θ (FewRel)	0.1
Encoder output size	768
BERT input max length	256
λ_1	1.0
λ_2 (FewRel)	1.0
λ_2 (TACRED)	2.0
λ_3	0.25
Soft prompt initialization	Random
Soft prompt phrase length	3
Soft prompt number of phrases	4

Table 4: Hyperparameters setting for LLMEs backbone.

Hyperparameter	Value
Encoder output size	4096
Epochs	10
Learning rate	1×10^{-5}, 1×10^{-4}
α	0.2
τ $\mathcal{L}_{\text{CMI}}$	0.05
τ $\mathcal{L}_{\text{WSC}}$	0.1
θ (FewRel)	0.3
θ (Tacred)	0.5
Lora alpha	16
Lora rank	8
Lora dropout	0.05
λ_1	1.0
λ_2	2.0
λ_3	0.5

Method	Tasks							
	$\mathcal{T}^1$	$\mathcal{T}^2$	$\mathcal{T}^3$	$\mathcal{T}^4$	$\mathcal{T}^5$	$\mathcal{T}^6$	$\mathcal{T}^7$	$\mathcal{T}^8$
FewRel *(10-way–5-shot)*								
SIRUS	$\mathbf{94.74}_{\pm 0.27}$	$\mathbf{87.12}_{\pm 2.21}$	$\mathbf{81.06}_{\pm 1.52}$	$\mathbf{77.49}_{\pm 2.58}$	$75.47_{\pm 2.6}$	$72.48_{\pm 1.75}$	$\mathbf{70.6}_{\pm 1.31}$	$\mathbf{69.16}_{\pm 0.43}$
w/o $\mathcal{L}_{WSC}$	$94.08_{\pm 0.44}$	$85.46_{\pm 2.23}$	$78.69_{\pm 1.72}$	$75.10_{\pm 2.71}$	$73.03_{\pm 2.76}$	$70.24_{\pm 1.78}$	$68.51_{\pm 1.23}$	$66.85_{\pm 0.77}$
w/o $\mathcal{L}_{CMI}$	$94.71_{\pm 0.38}$	$79.23_{\pm 3.41}$	$72.77_{\pm 1.01}$	$70.39_{\pm 1.98}$	$70.32_{\pm 1.94}$	$67.76_{\pm 1.74}$	$66.40_{\pm 1.44}$	$64.65_{\pm 1.22}$
w/o $\mathcal{L}_{DT}$	$94.69_{\pm 0.28}$	$86.68_{\pm 2.14}$	$80.71_{\pm 1.78}$	$77.13_{\pm 2.26}$	$\mathbf{75.52}_{\pm 2.66}$	$\mathbf{72.77}_{\pm 1.73}$	$70.41_{\pm 1.33}$	$68.29_{\pm 0.84}$
TACRED *(5-way-5-shot)*								
SIRUS	$87.41_{\pm 0.41}$	$84.28_{\pm 7.38}$	$\mathbf{76.38}_{\pm 3.99}$	$\mathbf{73.86}_{\pm 4.16}$	$\mathbf{68.06}_{\pm 5.57}$	$\mathbf{66.64}_{\pm 5.76}$	$\mathbf{62.74}_{\pm 3.92}$	$\mathbf{60.68}_{\pm 3.53}$
w/o $\mathcal{L}_{WSC}$	$\mathbf{87.79}_{\pm 0.41}$	$\mathbf{84.45}_{\pm 6.18}$	$75.91_{\pm 2.41}$	$73.67_{\pm 4.61}$	$67.76_{\pm 6.66}$	$66.35_{\pm 5.10}$	$61.55_{\pm 4.78}$	$59.48_{\pm 2.59}$
w/o $\mathcal{L}_{CMI}$	$86.49_{\pm 0.39}$	$79.98_{\pm 5.03}$	$71.2_{\pm 4.45}$	$66.25_{\pm 5.37}$	$62.47_{\pm 5.46}$	$61.23_{\pm 5.21}$	$56.08_{\pm 3.68}$	$54.30_{\pm 2.88}$
w/o $\mathcal{L}_{DT}$	$87.31_{\pm 0.54}$	$84.27_{\pm 6.69}$	$75.77_{\pm 4.66}$	$72.8_{\pm 4.27}$	$66.84_{\pm 4.53}$	$66.12_{\pm 5.84}$	$61.86_{\pm 3.96}$	$59.5_{\pm 3.66}$

Table 5: Ablation study (%) of loss functions. The best results are in **bold**.

limited contribution of $\mathcal{L}_{DT}$ to the model performance. One possible explanation is that, throughout the training process, we observe that the identified clusters exhibit stability and demonstrate minimal variation after a few optimization steps. This suggests that the model learns to cluster and identify similar classes effectively early on, with the support of all losses. As a result, which focuses on the relationship between samples and their cluster centroids, may contribute less during later training steps, where the emphasis shifts toward distinguishing samples between different relations. Nonetheless, the incorporation of still results in approximately a 1% improvement in model performance. Given already high accuracy and the challenging of FCRE, this incremental improvement is meaningful and highlights its effectiveness.

Influence of Clustering Algorithms: To assess the influence of different clustering algorithms in our method, we evaluate its performance using various clustering techniques. Specifically, we compare Agglomerative Clustering, K-means, and DBSCAN while employing BERT as the backbone model and conducting experiments on the TACRED dataset. The results, summarized in the Table 6, indicate that Agglomerative Clustering achieves superior performance, underscoring its effectiveness and suitability for our approach.

Cluster Algorithm	Accuracy on the final task
K-means with K = $\mathcal{R}$ / 2	66.89
K-means with K = $\mathcal{R}$ / 3	67.20
DBSCAN	67.18
Agglomerative Clustering	**69.16**

Table 6: Accuracy after training on the final task using different clustering algorithms. $\mathcal{R}$ represents the total number of relations in the dataset.

Hyper-parameter Sensitivity: To examine the impact of weighted parameters for each loss function, we conducted experiments by varying λ_2 and λ_3 while keeping $\lambda_1 = 1$ to limit the exponential growth of possible configurations. The results, presented in the Table 7 for the TACRED dataset using BERT backbone, exhibit low standard deviation across different parameter settings, suggesting minimal sensitivity to these hyperparameters. Notably, the lowest accuracy (57.16%) is observed when λ_2 and λ_3 are set to small values, emphasizing the critical role of each loss function in enhancing the model's performance.

	$\lambda_2 = 0.5$	$\lambda_2 = 1.0$	$\lambda_2 = 1.5$	$\lambda_2 = 2.0$
$\lambda_3 = 0.25$	57.16	59.37	59.57	60.68
$\lambda_3 = 0.5$	58.35	59.29	60.28	60.58
$\lambda_3 = 0.75$	58.94	58.84	59.94	60.44
$\lambda_3 = 1.0$	59.23	60.06	59.55	60.54

Table 7: Accuracy variations with different weighted parameter settings for each loss function.

Computational Overhead: To assess the computational efficiency of SIRUS, we compare its additional time cost against CPL and CPI. We measure the average training time per epoch on an RTX 3090 GPU using the TACRED dataset, with a fixed batch size of 16 and BERT as the backbone model. Compared to CPL (10.34s) and CPI_MI (11.30s), SIRUS (22.89s) introduces some additional computational overhead, primarily due to processing label descriptions and running the Agglomerative Clustering algorithm (L4-5 in Algorithm 1). However, the time spent on clustering is minimal, averaging only 0.05 seconds, with most of the overhead attributed to updating description embeddings via forward passes. However, in few-shot learning scenarios, the number of training samples is small, with only a few samples per class. As the training time of SIRUS is short, the trade-off between its time complexity and significant improvement in performance is acceptable. This trade-off enables SIRUS to outperform the two CPL-based models. On the other hand, CPL and CPL_MI methods rely on data augmentation techniques using LLMs to generate additional data, which increases computational complexity. The time cost for these methods can scale with the number of extra generated samples, further potentially adding overhead compared to the more efficient SIRUS approach.

C.2 Relation and Description

Table 8 provides details of the relations and their descriptions corresponding to each class index depicted in Figure 1. These descriptions provide clearer evidence of their similarity.

Index	Relation	Description
1	headquarters location	city where an organization's headquarters is or has been situated
2	work location	location where persons or organizations were actively participating in employment, business, or other work
3	location of formation	location where a group or organization was formed
4	located in the administrative territorial entity	the item is located on the territory of the following administrative entity
5	country of citizenship	the object is a country that recognizes the subject as its citizen
6	country	sovereign state of this item (not to be used for human beings)
7	country of origin	country of origin of this item (creative work, food, phrase, product, etc.)

Table 8: Corresponding relation and its description to class index in Figure 1.