Mitigating Non-Representative Prototypes and Representation Bias in Few-Shot Continual Relation Extraction

Thanh Duc Pham[‡§], **Nam Le Hai**[§*], **Linh Ngo Van**[§], **Diep Thi-Ngoc Nguyen**[♮], **Sang Dinh**[§], **Thien Huu Nguyen**[♭]

[‡]FPT Software AI Center, Vietnam, [§]Hanoi University of Science and Technology,
[♮]VNU University of Engineering and Technology, [♭]University of Oregon

[‡]thanhpd29@fpt.com, [§]{namlh, linhnv, sangdv}@soict.hust.edu.vn,
[♮]ngocdiep@vnu.edu.vn, [♭]thien@cs.uoregon.edu

Abstract

To address the phenomenon of similar classes, existing methods in few-shot continual relation extraction (FCRE) face two main challenges: non-representative prototypes and representation bias, especially when the number of available samples is limited. In our work, we propose Minion to address these challenges. Firstly, we leverage the General Orthogonal Frame (GOF) structure, based on the concept of Neural Collapse, to create robust class prototypes with clear separation, even between analogous classes. Secondly, we utilize label description representations as global class representatives within the fast-slow contrastive learning paradigm. These representations consistently encapsulate the essential attributes of each relation, acting as global information that helps mitigate overfitting and reduces representation bias caused by the limited local few-shot examples within a class. Extensive experiments on well-known FCRE benchmarks show that our method outperforms state-of-the-art approaches, demonstrating its effectiveness for advancing RE system.

1 Introduction

Few-Shot Continual Relation Extraction (FCRE) has emerged as a significant research focus due to its critical role in addressing the challenges of adapting to evolving relations with limited data (Qin and Joty, 2022; Chen et al., 2023) for practical RE systems. Similar to other continual learning (CL) systems, FCRE faces the issue of catastrophic forgetting (Thrun and Mitchell, 1995), where previously acquired knowledge degrades as new tasks are learned. Recent FCRE studies (Wang et al., 2023; Ma et al., 2024; Luo et al., 2024; Tran et al., 2024; Nguyen et al., 2025b) explored strategies for storing old samples to preserve critical attributes of relations, ensuring that the representations of all relations across tasks remain distinguishable.

The presence of analogous classes has been recognized as a key factor contributing to catastrophic forgetting in continual relation extraction (Wang et al., 2022; Nguyen et al., 2023b; Le et al., 2024c, 2025b). In the context of FCRE, this phenomenon can be more severe due to the limited data, which constrains the model's ability to effectively distinguish between similar classes Wang et al. (2022). However, this topic remains largely underexplored within the FCRE scenario. In our work, we tackle two critical challenges faced by existing state-of-the-art FCRE methods: non-representative prototypes and representation bias.

First, existing FCRE models commonly employ a prototype mechanism, where a class prototype is represented by averaging the embeddings of all samples within that class, and these prototypes are then used to predict relations for test samples. When the number of available samples is limited, this mechanism may generate *non-representative prototypes* (Li and Lyu, 2024) and does not guarantee the ability to distinguish between analogous classes effectively. Recently, Papyan et al. (2020) reveals a famous phenomenon of deep learning models, refer as neural collapse (NC), that: on a balanced dataset, at the terminal phase of training (training error equals 0), the last-layer feature of the same class will converge to their within-class means; the class classifiers and their corresponding within-class mean will be formed as a simplex equiangular tight frame (ETF). Inspired by Papyan et al. (2020), Yang et al. (2023), and Yang et al. (2022) have initialized the final class classifiers (prototypes) using the ETF structure, keeping them fixed during continual training. These models then align the last layer features of input samples with their corresponding prototypes. This approach has demonstrated competitive performance in both joint training and Few-shot Class Incremental Learning (FSCIL). However, these approaches require initializing the ETF structure with

*Corresponding author: namlh@soict.hust.edu.vn

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10791–10809

a fixed number of classes, K, where each vertex of the ETF has the same pairwise angle of $(-\frac{1}{K-1})$. While Yang et al. (2023) suggested that a large K could be set relative to the number of classes in a specific dataset, this constraint renders the model impractical for real-world applications and is not sufficient to the objectives of continual learning, where the number of classes is typically unknown and dynamically evolving.

Second, Song et al. (2023) introduced the concept of *representation bias*, which refers to a phenomenon where the representations learned by a model are locally sufficient for the current task but may prove globally inadequate for classifying analogous classes in future tasks. This arises as CL models often discard features irrelevant to the present task, which may later be critical. They proposed InfoCL to improve the CL model's capacity to learn more comprehensive and robust representations by introducing *fast-slow* and *current-past* contrastive losses. However, several factors may limit the effectiveness of InfoCL in the context of FCRE. While InfoCL aims to capture a wide range of features, including those irrelevant to the current task, to benefit future tasks such as distinguishing analogous classes, the limited number of data samples in the few-shot scenario poses a significant challenge in effectively representing distinctions between similar classes. Moreover, relying solely on contrastive learning with the sample itself may be insufficient for acquiring globally informative representations. The limited data in FCRE tasks restricts the ability to fully capture relational nuances (Wang et al., 2023; Li and Lyu, 2024), emphasizing the necessity of incorporating robust and comprehensive global context to enrich the representation and improve the information compression.

In our work, we propose Minion, a novel and universal approach to **mi**tigate **n**on-representat**i**ve prototypes and representati**on** bias in FCRE. Firstly, ProtoGOF is introduced in Minion, leveraging the General Orthogonal Frame (GOF) structure (Dang et al., 2023) instead of ETF to create class prototypes. This approach enables the construction of robust class prototypes and aligns the final representations of input sentences to the GOF through contrastive learning. Unlike methods reliant on predefined class limits, such as the ETF structure, our framework is well-suited for continual learning settings, providing flexibility without requiring prior knowledge of the maximum number of classes. Furthermore, Minion ensures the creation of robust class prototypes with clear separation, even between analogous classes. Secondly, the Fast-Slow mechanism (He et al., 2020) in Minion is enhanced through the integration of label description representations, termed *Fast-Slow Contrastive Learning with Label Descriptions* (FCLD). FCLD incorporates label description representations to consistently encapsulate the essential attributes of each relation, acting as global information that helps mitigate overfitting and reduces representation bias caused by the limited local few-shot examples within a class. Therefore, Minion not only improves the model's ability to learn comprehensive and robust representations but also establishes a more effective mechanism for distinguishing analogous classes to reduce catastrophic forgetting. Extensive experiments conducted on two FCRE benchmarks, TACRED and FewRel, reveal that our method surpasses state-of-the-art approaches, representing a significant advancement in FCRE research.

2 Background

2.1 Problem Formulation

Few-Shot Continual Relation Extraction (FCRE) represents a formidable challenge within natural language processing, as it integrates the difficulties of continual learning with the inherent limitations of few-shot scenarios. A detailed discussion of related works can be found in Appendix A. In the FCRE framework, a model is sequentially exposed to a series of tasks, $\mathcal{T} = \{\mathcal{T}^1, \mathcal{T}^2, ..., \mathcal{T}^n\}$, where each task introduces a new set of relations, R_i, to be learned. For a given task $\mathcal{T}^i$, the model is provided with a limited dataset $\mathcal{D}_i = \{(x_j, r_j)\}_{j=1}^{m}$, where $m = N \times K$ defines the total number of examples, with N representing the number of new relations and K denoting the few-shot samples per relation. Each example comprises 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 configuration aligns with the *"N-way-K-shot"* learning paradigm, as described by Chen et al. (2023).

The core challenge of FCRE involves balancing two closely connected objectives: enabling the model to efficiently learn newly introduced relations from a limited number of examples (few-shot learning) while concurrently preserving knowledge of previously acquired relations (continual learning). Successfully addressing this challenge ne-

cessitates effective regulation of the model's capacity for adaptability to new information (plasticity) and its ability to maintain existing knowledge (stability). During the evaluation phase, the model's performance is assessed using a comprehensive test set, $\mathcal{D}^{test}$, encompassing all relations $R_{total} = \bigcup_{i=1}^{n} R_i$ encountered across tasks. This evaluation assesses the model's capability to learn newly introduced relations while preserving its proficiency in previously learned ones. The FCRE formulation underscores the importance of developing adaptive, efficient systems for relation extraction in dynamic, low-resource settings.

2.2 Input Formulation and Representation

In Relation Extraction (RE), a common deep learning approach (Ji et al., 2020; Wang and Lu, 2020) leverages pre-trained language models (PLMs) like BERT (Devlin et al., 2019) to encode input data. Effective input formulation is critical for obtaining high-quality embeddings for classification. Early methods often rely on the `[CLS]` token concatenated with the input x and use its vector representation for classification. Alternatively, some approaches incorporate special tokens to highlight the two entities and concatenate their embeddings as input to the relation classification layer (Zhao et al., 2022; Le et al., 2024c).

In this study, we adopt the input format proposed by Ma et al. (2024), where a special `[MASK]` token represents the relation between the head entity (e_h) and tail entity (e_t). This token is combined with the original sentence x and the two entities. Additionally, learnable tokens are incorporated to reduce dependence on handcrafted tokens, resulting in the following input formulation:

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

where $[h_i]$ 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 $[h]$. 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.

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

where $f_{\mathcal{M}}(X)$ denotes the forward function of an encoder $\mathcal{M}$ on input X. The latent representation is subsequently utilized for learning and to predict the relation corresponding to the given input x.

2.3 Neural Collapse

Papyan et al. (2020) reveal the neural collapse phenomenon, that the last-layer features converge to their within-class means, and the within-class means together with the classifier vectors collapse to the vertices of a simplex equiangular tight frame at the terminal phase of training on a balanced dataset.

Definition 2.1 (Simplex Equiangular Tight Frame)**.** A collection of vectors $\mathbf{m}_i \in \mathbb{R}^d$, $i = 1, 2, \cdots, K$, $d \geq K-1$, is said to be a simplex equiangular tight frame if:

$$\mathbf{M} = \sqrt{\frac{K}{K-1}}\mathbf{U}\left(\mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^T\right), \tag{3}$$

where $\mathbf{M} = [\mathbf{m}_1, \cdots, \mathbf{m}_K] \in \mathbb{R}^{d\times K}$, $\mathbf{U} \in \mathbb{R}^{d\times K}$ allows a rotation and satisfies $\mathbf{U}^T\mathbf{U} = \mathbf{I}_K$, $\mathbf{I}_K$ is the identity matrix, and $\mathbf{1}_K$ is an all-ones vector.

All vectors in a simplex ETF have an equal ℓ_2 norm and the same pair-wise angle, *i.e.,*

$$\mathbf{m}_i^T\mathbf{m}_j = \frac{K}{K-1}\delta_{i,j} - \frac{1}{K-1}, \forall i, j \in [1, K],$$

where $\delta_{i,j}$ equals 1 when $i = j$ and 0 otherwise. The pair-wise angle $-\frac{1}{K-1}$ is the maximal equiangular separation of K vectors in $\mathbb{R}^d$.

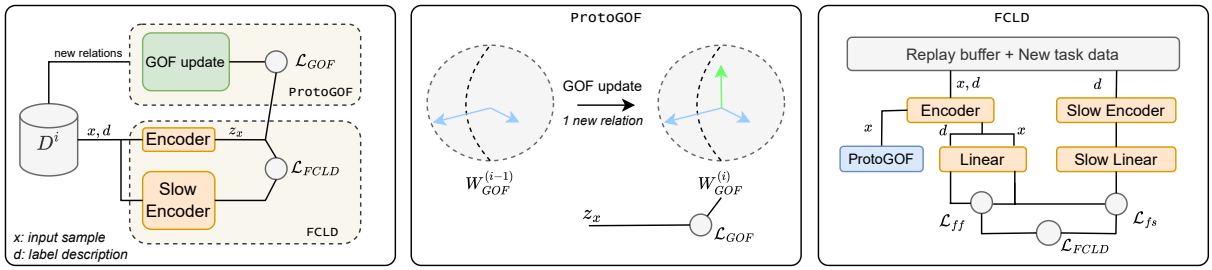
Then the neural collapse (NC) phenomenon can be formally described as:

(NC1) Within-class variability of the last-layer features collapse: $\Sigma_W \to \mathbf{0}$, and $\Sigma_W := \text{Avg}_{i,k}\{(z_{k,i} - z_k)(z_{k,i} - z_k)^T\}$, where $z_{k,i}$ is the last-layer feature of the i-th sample in the k-th class, and $z_k = \text{Avg}_i\{z_{k,i}\}$ is the within-class mean of the last-layer features in the k-th class;

(NC2) Convergence to a simplex ETF: $\tilde{z}_k = (z_k - z_G)/||z_k - z_G||, k \in [1, K]$, satisfies Eq. (4), where z_G is the global mean of the last-layer features, *i.e.,* $z_G = \text{Avg}_{i,k}\{z_{k,i}\}$;

(NC3) Self duality: $\tilde{z}_k = \mathbf{w}_k/||\mathbf{w}_k||$, where $\mathbf{w}_k$ is the classifier vector of the k-th class;

(NC4) Simplification to the nearest class center prediction: $\arg\max_k \langle z, \mathbf{w}_k\rangle = \arg\min_k ||z - z_k||$, where z is the last-layer feature of a sample to predict for classification.



(a) (b) (c)

Figure 1: An overview of our Minion framework: ***(a)*** showcases the data flow through its two primary components: ProtoGOF and FCLD, ***(b)*** illustrates the adoption of GOF to construct relational prototypes in the development of ProtoGOF, as detailed in Section 3.1 and ***(c)*** depicts forward flow of FCLD, as discussed in Section 3.2. x and d illustrate the forward paths of the input data and its corresponding label description, respectively.

Definition 2.2 (General Orthogonal Frame)**.** A standard general orthogonal frame (GOF) is a collection of points in $\mathbb{R}^K$ specified by the columns:

$$\mathbf{N} = \frac{1}{\sqrt{\sum_{k=1}^{K} a_k^2}} \operatorname{diag}(a_1, a_2, \ldots, a_K), \quad (4)$$

where $a_i > 0, \forall\, i \in [K]$ denotes the number of samples associated with the corresponding class in the data set. Dang et al. (2023) also examined the general version of GOF as a collection of points in $\mathbb{R}^d$ $(d \geq K)$ specified by the columns of $\mathbf{PN}$ where $\mathbf{P} \in \mathbb{R}^{d \times K}$ is an orthonormal matrix, i.e. $\mathbf{P}^\top \mathbf{P} = \mathbf{I}_K$.

However, ETF is observed exclusively in balanced datasets, limiting its applicability to most real-world datasets, which are often inherently imbalanced. Dang et al. (2023) demonstrated that the final-layer features and classifiers converge to a geometric structure known as GOF (Definition 2.2), characterized by orthogonal vectors with lengths proportional to the amount of data in their respective classes.

3 Methodology

This section details two core components of Minion: ProtoGOF and FCLD. The overall framework pipeline is illustrated in Figure 1a.

3.1 Guiding Class Prototypes through General Orthogonal Frame

In this section, we present our proposed mechanism, ProtoGOF, which utilizes the General Orthogonal Frame structure to construct class prototypes and guide relation representations towards an NC solution with distinct separation. We demonstrate the advantages of ProtoGOF compared to existing studies, highlighting its suitability for CL settings and its adaptability to a wide range of dataset distribution scenarios.

Assigning fixed class prototypes based on NC (Song et al., 2023; Yang et al., 2022, 2023) has demonstrated remarkable performance in continual learning for classification tasks, including fine-grained and few-shot settings. However, existing approaches have not addressed the potential impact of imbalanced training conditions on the structure of traditional NC (i.e. ETF). Furthermore, Yang et al. (2023) necessitate the pre-assignment of a fixed number of classes, K, during the initialization of prototypes, as each prototype in the ETF structure must maintain the same pairwise angle $(-\frac{1}{K-1})$ to ensure maximal separation. Although setting K to a large value can help mitigate the risk of missing relations during the learning process, this constraint limits the structure's ability to achieve optimal class separation. Additionally, it restricts the flexibility needed to accommodate a number of relations that may exceed the predefined value of K, which conflicts with the goals of continual learning frameworks.

On the other hand, Dang et al. (2023) emphasized that in imbalanced scenarios, the representations of the last-layer features and classifiers converge to the GOF structure. Therefore, we explore an innovative approach that employs GOF as a fixed structure to address the challenge of dynamic class numbers, while guiding relation representations toward effective separation. As GOF necessitates an orthogonal arrangement where the angles between prototypes are independent of the number of classes, it facilitates the incorporation of emerging class prototypes corresponding to new relations without compromising any of GOF's characteristic.

Moreover, the class prototypes are orthogonal to each other, ensuring effective separation between classes and improving the model's capacity to differentiate analogous classes.

In particular, during the i^{th} task, we adopt the GOF structure (depited in Figure 1b) with following steps:

- **Step 1:** If $i = 0$, the class prototypes (W^0_{GOF}) are initialized according to the GOF structure described in Definition 2.2, corresponding to the number of relations and their respective sample counts in $\mathcal{D}_0$.
- **Step 2:** Update the GOF structure $W^{(i)}_{GOF}$: Prototypes for newly emerging classes from $\mathcal{D}_i$ are initialized and integrated into $W^{(i-1)}_{GOF}$. These prototypes are constructed to ensure that the updated structure, $W^{(i)}_{GOF}$, satisfies the conditions of the GOF framework.
- **Step 3:** Building on these prototypes, a loss function is designed to align input sentence representations with their respective class prototypes while distancing them from prototypes of other classes (W_{GOF^-}). Follow the ProxyNCA loss introduced by Movshovitz-Attias et al. (2017), we define $\mathcal{L}_{GOF}$ as follows:

$$\begin{aligned}\mathcal{L}_{GOF} &= \sum_{i=1}^{n} \sum_{\mathbf{x}\in\mathcal{D}_i} -\log \frac{e^{d(\mathbf{z}_x, w_j^+)}}{\sum\limits_{w_j^- \in W_{GOF^-}} e^{d(\mathbf{z}_x, w_j^-)}} \\ &= \sum_{i=1}^{n} \sum_{x\in\mathcal{D}_i} \left\{ -d(\mathbf{z}_x, w_j^+) + \underset{w_j^- \in W_{GOF^-}}{\text{LSE}} \; d(\mathbf{z}_x, w_j^-) \right\}, \end{aligned} \tag{5}$$

where $\mathbf{z_x}$ is the representation of input sentence calculated as in the Eq (2); W_{GOF^-} is set of all negative proxies (set of prototypes w.r.t different classes from the class of x), w_j^+ and w_j^- respectively denote positive and negative prototypes; LSE stands for LogSumExp loss.

3.2 Fast-Slow Contrastive Learning with Label Descriptions

Representation bias, as introduced by Song et al. (2023), refers to a phenomenon observed in CL scenarios (discussed further in Appendix B). It highlights that during the i^{th} task, compressing the essential representations of current relations may lead to insufficient global information. To mitigate this issue, InfoCL, including the Fast-Slow Contrastive method has been proposed, which employs two distinct encoders: a fast encoder and a slow momentum-updated encoder. These encoders are designed to capture comprehensive information across different optimization stages: the later stages compress essential generalization (fast encoder), while the early optimization phase preserves more detailed input information (slow encoder). While this approach proves effective in settings with rich data, its performance in few-shot scenarios could be challenging due to the limited representativeness of the available samples. These constraints result in inconsistent representations, as the sparse data often fails to encapsulate the critical attributes of their respective relations (Han et al., 2021; Wang et al., 2023; Li and Lyu, 2024), potentially leading to global insufficiency in distinguishing classes. Label descriptions have demonstrated their ability to enhance the performance of prototypical networks for few-shot RE by providing supplementary insights into relation types (Yang et al., 2020; Liu et al., 2022; Luo et al., 2024; Borchert et al., 2024). Thus, these descriptions potentially offer a consistent, global class-specific context, delivering more reliable global information. This ensures that representations can capture the essential characteristics of relations, thereby mitigating the risk of incorporating misleading information.

From this perspective, we introduce Fast-Slow Contrastive Learning with Label Descriptions (FCLD), which leverages the detailed information provided by relational descriptions and transfers this knowledge to input samples through contrastive learning, thereby enhancing the model's representation capacity. Specifically, instead of using input sentences, we utilize label descriptions as inputs to the slow encoder. The resulting representations are then incorporated as distilled information to enhance the input representations generated by the fast encoder for classification. Furthermore, the slow encoder, utilizing label descriptions, captures information closely aligned with the descriptions, while the essential information compressed by the fast encoder remains valuable. Therefore, we additionally leverage the representations of label descriptions generated by the fast encoder to enrich the overall information. Figure 1c provides an illustration of the FCLD.

We employ the InfoNCE loss (van den Oord et al., 2019) to preserve information from the label descriptions processed by the slow branch to the

input representations in the fast branch:

$$\mathcal{L}_{fs} = -\frac{1}{|B|}\sum_{i=1}^{|B|}\log\frac{\exp(\mathbf{z}_{\mathbf{x_i}}\cdot\widetilde{\mathbf{d}^{\mathbf{s}}_{\mathbf{x_i}}}/\tau)}{\sum_{j=1}^{|B|}\exp(\mathbf{z}_{\mathbf{x_i}}\cdot\widetilde{\mathbf{d}^{\mathbf{s}}_{\mathbf{x_j}}}/\tau)}, \tag{6}$$

where B denotes a batch with size $|B|$ during training, $\widetilde{\mathbf{d}^{\mathbf{s}}_{\mathbf{x}}}$ and $\mathbf{z}_{\mathbf{x_i}}$ respectively represent the label description embedding generated by the slow model and the feature representation of input x_i from the fast model, τ is the temperature. Similarly, the label description information derived from the fast encoder is transferred to the input through Equation 6 ($\mathcal{L}_{ff}$), utilizing the description representation $\mathbf{d}^{\mathbf{f}}_{\mathbf{x}}$ generated by the fast encoder. Finally, the loss $\mathcal{L}_{FCLD}$ is defined as:

$$\mathcal{L}_{FCLD} = \lambda_1\mathcal{L}_{fs} + \lambda_2\mathcal{L}_{ff} \tag{7}$$

where λ_1 and λ_2 are weighted hyper-parameters.

3.3 Training and Inference Procedures

Training Procedure: Algorithm 1 introduces the end-to-end training procedure for each task $\mathcal{T}_j$, wherein Φ^F_{j-1}, Φ^S_{j-1} denote the parameters of fast and slow encoders after training in the previous $j-1$ task. Following existing memory-based methods, we maintain a memory buffer $\tilde{M}_{j-1}$ that stores a limited number of representative samples from all previous tasks $\mathcal{T}_1, \ldots, \mathcal{T}_{j-1}$, and a set $\tilde{Des}_{j-1}$ consisting of descriptions (d_i) w.r.t. all previously encountered relations (r_i).

1. **Initialization** (Line 1): The parameters of the fast and slow models for the current task, Φ^F_j and Φ^S_j, are initialized using the parameters of the models from the previous task $(i-1)^{th}$ and the relation set $\tilde{R}_j$ and the relation description set $\tilde{Des}_j$ are updated to incorporate the newly introduced relations.

2. **GOF Structure Updating** (Line 2): The model updates the GOF structure by incorporating prototypes for the newly emerging classes from D^i_{train}, following steps 1 and 2 outlined in Section 3.1.

3. **Model Training** (Lines 3–8): The current data is combined with the replay data to fine-tune the model using a unified loss function. The current data is combined with the replay data to fine-tune the model using a unified loss function. Specifically, $\mathcal{L}_{GOF}$ and $\mathcal{L}_{FCLD}$ are computed as defined in Equations 5 and 7, while $\mathcal{L}_{SCL}$, a supervised contrastive loss commonly employed in prior studies (Khosla et al., 2021; Ma et al., 2024; Cui et al., 2021), ensures separation of samples from different classes in the representation space.

4. **Update Replay Buffer** (Lines 9–13): We select L samples from D^{train}_j for each relation $r_i \in R_j$, choosing those whose latent representations are closest to the class centroid.

Our training objective function is as below:

$$\mathcal{L} = \alpha\mathcal{L}_{SCL} + \mathcal{L}_{FCLD} + \lambda_3\mathcal{L}_{GOF} \tag{8}$$

where α and λ_3 are weighted hyper-parameters.

Inference Procedure: Given a sample x with hidden representation z_x and a set of relation prototypes W_{GOF} that follow the GOF structure, the relation is predicted by measuring the similarity between the input x and the prototypes, as follows:

$$r^* = \underset{r}{\text{argmax}}\,(\gamma(\boldsymbol{z}_x, \boldsymbol{w}_r)) \tag{9}$$

where $w_r \in W_{GOF}$ and $\gamma(\cdot,\cdot)$ denotes the cosine similarity function.

4 Experimental Results

4.1 Experiment Setup

We conduct evaluation on two benchmark datasets FewRel (Han et al., 2018) and TACRED (Zhang et al., 2017) against 8 state-of-the-art (SOTA) baselines. In addition to BERT (Devlin et al., 2019), we also incorporate LLM2Vec variants (BehnamGhader et al., 2024), which are large language models (LLMs) adapted to serve as text encoders with improved representation capabilities, rather than as generative decoders. These are referred to as Large Language Model Embeddings (LLMEs), with further details provided in Appendix C.3. Since its functionality and usage are identical to the BERT encoder, it can be easily integrated into a wide range of frameworks in contrast to the use of causal LLMs in CPL_MI (Tran et al., 2024). Specifically, we evaluate our approach using three LLM2Vec variants—LLama2, LLama3, and Mistral—as backbone models. For each task, we report performance on the updated $\mathcal{D}^{test}$, presenting both the mean and standard deviation of accuracy across six random seeds. Additional experiments are provided in the Appendix C.

Backbone	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)*									
BERT	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)	**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)	94.87	85.14	78.80	75.10	72.57	69.57	66.85	64.50
	CPL_MI (Tran et al., 2024)	94.69$_{\pm 0.70}$	85.58$_{\pm 1.88}$	80.12$_{\pm 2.45}$	75.71$_{\pm 2.28}$	73.90$_{\pm 1.80}$	70.72$_{\pm 0.91}$	68.42$_{\pm 1.77}$	66.27$_{\pm 1.58}$
	CPL+MI+augment(Anh et al., 2025)	**94.76**	85.48	80.24	77.69	75.60	72.94	70.74	68.36
	SIRUS (Le et al., 2025a)	94.74$_{\pm 0.27}$	**87.12$_{\pm 2.21}$**	**81.06$_{\pm 1.52}$**	77.49$_{\pm 2.58}$	75.47$_{\pm 2.60}$	72.48$_{\pm 1.75}$	70.60$_{\pm 1.31}$	69.16$_{\pm 0.43}$
	Minion	94.58$_{\pm 0.36}$	86.96$_{\pm 2.56}$	81.02$_{\pm 1.33}$	**77.66$_{\pm 2.18}$**	**75.66$_{\pm 1.96}$**	**73.05$_{\pm 1.25}$**	**71.33$_{\pm 1.16}$**	**69.61$_{\pm 2.12}$** ↑ 0.45
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†	**96.10$_{\pm 0.25}$**	87.92$_{\pm 2.03}$	81.98$_{\pm 2.91}$	79.91$_{\pm 3.29}$	77.81$_{\pm 3.25}$	75.56$_{\pm 1.49}$	73.90$_{\pm 2.33}$	72.27$_{\pm 1.98}$
	CPL	95.82$_{\pm 0.30}$	86.45$_{\pm 1.84}$	82.10$_{\pm 2.02}$	80.11$_{\pm 1.47}$	77.64$_{\pm 3.17}$	76.05$_{\pm 1.62}$	74.60$_{\pm 2.77}$	72.81$_{\pm 1.79}$
	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$_{\pm 0.50}$
	Minion	95.25 ± 0.44	**88.12 ± 2.92**	**84.34 ± 1.26**	**82.31 ± 1.24**	**81.49 ± 1.92**	79.02 ± 1.43	**78.03 ± 0.98**	**76.27 ± 0.81** ↑ 0.29
LLama3	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$_{\pm 1.11}$
	CPL+MI+augment	**94.76**	85.48	80.24	77.69	75.60	72.94	70.74	68.36
	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$_{\pm 0.98}$
	Minion	95.97$_{\pm 0.48}$	**91.38$_{\pm 3.27}$**	**88.33$_{\pm 2.19}$**	**86.74$_{\pm 1.75}$**	**85.8$_{\pm 2.96}$**	**84.56$_{\pm 2.38}$**	**82.91$_{\pm 1.75}$**	**81.01$_{\pm 0.53}$** ↑ 2.19
Mistral	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$_{\pm 1.32}$
	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$_{\pm 1.15}$**
	Minion	95.95$_{\pm 0.28}$	**90.78$_{\pm 2.56}$**	**87.31$_{\pm 2.71}$**	84.08$_{\pm 2.43}$	**82.58$_{\pm 2.74}$**	**80.09$_{\pm 2.19}$**	**78.64$_{\pm 1.40}$**	76.61$_{\pm 0.61}$
TACRED *(5-way-5-shot)*									
BERT	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)	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)	**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$_{\pm 0.80}$	82.54$_{\pm 2.98}$	75.12$_{\pm 3.67}$	70.65$_{\pm 2.75}$	66.79$_{\pm 2.18}$	65.17$_{\pm 2.48}$	61.25$_{\pm 1.52}$	59.48$_{\pm 3.53}$
	CPL+MI+augment (Anh et al., 2025)	86.33	82.31	**76.35**	70.93	**68.28**	65.04	62.60	**61.97**
	SIRUS (Le et al., 2025a)	87.41$_{\pm 0.41}$	**84.28$_{\pm 7.38}$**	76.38$_{\pm 3.99}$	**73.86$_{\pm 4.16}$**	68.06$_{\pm 5.57}$	66.64$_{\pm 5.76}$	62.74$_{\pm 3.92}$	60.68$_{\pm 3.53}$
	Minion	87.28$_{\pm 0.23}$	81.89$_{\pm 1.35}$	75.72$_{\pm 4.6}$	72.75$_{\pm 3.94}$	66.68$_{\pm 5.27}$	**66.71$_{\pm 6.6}$**	**62.81$_{\pm 5.55}$**	61.11$_{\pm 2.58}$
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$_{\pm 5.02}$
	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$_{\pm 0.59}$
	Minion	**89.58$_{\pm 0.36}$**	86.67$_{\pm 6.64}$	**79.91$_{\pm 4.75}$**	**77.34$_{\pm 2.59}$**	**75.91$_{\pm 3.62}$**	**74.78$_{\pm 2.07}$**	**72.75$_{\pm 2.15}$**	**72.35$_{\pm 1.03}$** ↑ 1.47
LLama3	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$_{\pm 3.71}$
	Minion	88.11$_{\pm 1.03}$	**86.82$_{\pm 3.67}$**	**83.13$_{\pm 3.08}$**	**78.29$_{\pm 4.25}$**	**77.19$_{\pm 4.15}$**	**76.38$_{\pm 3.87}$**	**74.84$_{\pm 3.38}$**	**74.49$_{\pm 2.71}$** ↑ 0.52
Mistral	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$_{\pm 1.97}$
	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$_{\pm 1.40}$	73.64$_{\pm 4.97}$	**73.06$_{\pm 3.23}$**
	Minion	87.56$_{\pm 0.55}$	**85.09$_{\pm 4.84}$**	**80.51$_{\pm 4.13}$**	**77.33$_{\pm 3.06}$**	**75.93$_{\pm 4.57}$**	**75.62$_{\pm 2.46}$**	**73.56$_{\pm 6.31}$**	72.47$_{\pm 3.60}$

Table 1: Accuracy (%) of existing methods using BERT-based, LLM and LLME-based backbones after training for each task. † denotes the use of original LLMs with causal mask. The best results are in **bold**, while the second-highest scores are underlined. All the baseline results are extracted from Wang et al. (2023) and Tran et al. (2024). ConPL§ results that are reproduced with the same settings as other models (Appendix C.2). The baseline results of original LLMs with the causal mask are obtained from Tran et al. (2024).

4.2 Main Results

In this section, we present the main results of our experiments, highlighting the performance of Minion compared to state-of-the-art baselines and its adaptability across various model architectures.

Minion versus SOTA baselines: To ensure a fair comparison with prior studies, we adopt BERT as the baseline model and utilize a replay buffer to retain a limited selection of representative samples. In particular, Table 1 demonstrates that Minion consistently achieves significant improvements across both benchmarks. Specifically, Minion outperforms the second-best method (SIRUS) on the FewRel dataset in 5 out of 8 sessions during the sequential training process and achieves 0.45% higher accuracy after the final task. A similar trend is observed on the TACRED dataset, where our Minion outperforms SIRUS by approximately 0.42% and consistently exceeds all baseline methods across the 8-task sessions.

Adaptability of Minion: While Minion demonstrates significant performance improvements over baselines with BERT, concerns may arise regarding the versatility of our framework when applied to large-scale models. To showcase the adaptability of Minion, we conduct experiments using LLMEs. The effectiveness of LLMEs compared to BERT is evident, with Minion improving performance by

Algorithm 1 Training procedure at each task T_j

Input:
$\mathcal{D}_j^{train}$: training data of task $\mathcal{T}^j$.
$\mathcal{D}^{test}$: Test data of all seen tasks.
$W_{GOF_{(j-1)}}$: GOF structure of the $(j-1)^{\text{th}}$ task.
L: The number of training samples allocated to memory for each relation.
Fast and Slow encoder parameters: $\Phi_{j-1}^F, \Phi_{j-1}^S$.
Previous variables: $\tilde{R}_{j-1}, \tilde{M}_{j-1}, \tilde{Des}_{j-1}$.
Current variables: $D_j^{train}, D_j^{test}, R_j, Des_j$.

Output:
$\Phi_j^F, \Phi_j^S, \tilde{M}_j, \tilde{R}_j, \tilde{Des}_j, W_{GOF_i}$.

Initialization: $\Phi_j^F, \Phi_j^S \leftarrow \Phi_{j-1}^F, \Phi_{j-1}^S$, $\tilde{R}_j \leftarrow \tilde{R}_{j-1} \cup R_j$, $\tilde{Des}_j \leftarrow \tilde{Des}_{j-1} \cup Des_j$
Updating W_{GOF_i} using $W_{GOF_{i-1}}$ and D_j^{train}
for batch in batches($\tilde{M}_{j-1} \cup D_j^{train}$) **do**
    $\widetilde{\mathbf{d^s_{x_i}}} \leftarrow f_{\mathcal{M}_{\Phi_j^S}}(d_i), \widetilde{\mathbf{d^f_{x_i}}} \leftarrow f_{\mathcal{M}_{\Phi_j^F}}(d_i)$
    Compute $\mathcal{L}$ as in Eq. 8
    Update Φ_j^F
    Update Φ_j^S
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
$\mathcal{D}^{test} \leftarrow \mathcal{D}^{test} \cup D_j^{test}$ ▷ Update test set

over 10% in both settings. Besides, LLMEs also clearly outperform causal LLMs, as their adaptation enables CPL to surpass CPL_MI with causal LLMs by around 1% using LLama2 backbone. Minion further demonstrates a remarkable improvement of 3-4% over CPL_MI and SIRUS with LLMs on both benchmarks, with LLama3 achieving the highest performance across all scenarios. These results demonstrate the adaptability and scalability of our approach across a wide range of architectures.

4.3 Ablation Study

In this section, we emphasize the key contributions of ProtoGOF and FCLD by systematically removing each component from the framework. Additionally, we aim to demonstrate the superiority of adopting the GOF structure over the ETF structure used in previous works. The complete results for each task are presented in Table 6 in Appendix D.2.

Method	FewRel	TACRED
Minion	**69.61**	**61.11**
w $ProtoETF$	68.60	60.28
w/o $ProtoGOF$	68.27	59.66
w/o $FCLD$	68.32	59.49
InfoCL (Song et al., 2023)	65.72	51.02

Table 2: The results of the ablation study (%) when removing each component of Minion on the final task using the BERT backbone. ProtoETF refers to the integration of the ETF structure into our framework as a replacement for GOF. InfoCL is reimplemented for FCRE without incorporating label descriptions in the fast-slow contrastive learning.

Effectiveness of ProtoGOF: As shown in Table 2, replacing the GOF structure with the ETF results in a performance drop of approximately 1% in accuracy. This underscores the limitation of the ETF structure, as models fail to achieve this structure due to the dynamic nature of class numbers and the imbalance present in the scenario. Besides, removing this component from Minion leads to a significant performance drop of nearly 2%, emphasizing the critical role of ProtoGOF in our framework. Additionally, using only ProtoGOF (w/o $FCLD$) results in performance comparable to using only FCLD, suggesting that this mechanism also helps address the issue of analogous classes. A more detailed discussion can be found in Appendix D.1, which demonstrates the reduction of confusion in predictions between similar classes when incorporating ProtoGOF.

Effectiveness of FCLD: The experimental results further highlight the importance of FCLD in our framework, as removing this component leads to a 2% performance drop on both benchmarks. In addition, to demonstrate the effectiveness of using label descriptions, we compare the model with FCLD (w/o $ProtoGOF$ in Table 2) and implement InfoCL for FCRE. The results show a substantial performance drop of around 3% on FewRel and over 8% on TACRED, highlighting the valuable role of label descriptions in FCLD for enhancing input representations and addressing the issue of analogous classes.

Computational Overhead: To illustrate the computational efficiency of each proposed component in Minion, we present the additional time cost in the Table 3. This comparison evaluates the training times of three methods for one epoch, using the

	Minion	w/o GOF	w/o FCLD	w/o GOF+FCLD
TACRED				
Avg training time / 1 epoch (s)	65.58	65.25	43.60	43.54
FewRel				
Avg training time / 1 epoch (s)	109.10	108.93	77.65	77.59

Table 3: Average training time per epoch under different model ablations.

same batch size of 16 and BERT as the backbone model on a Tesla P100-PCIE with16GB VRAM. The results indicate that Minion introduces some additional computational overhead, with FCLD being more computationally expensive than ProtoGOF due to the additional forward pass through the two encoders (as described in L4 of Algorithm 1). However, this overhead is minimal, adding only approximately over 20 seconds per epoch. Additionally, the small number of samples per class in few-shot settings further mitigates the impact of the overhead. Given that the training time for Minion is relatively short, the trade-off between the slight increase in computational overhead and the significant improvement in performance is acceptable.

5 Conclusion

In conclusion, we present Minion, a novel and versatile framework designed to address the challenges of non-representative prototypes and representation bias in FCRE. By introducing ProtoGOF, which leverages the General Orthogonal Frame (GOF) structure, our approach constructs robust class prototypes without relying on predefined class limits, making it well-suited for continual learning scenarios. Additionally, our Fast-Slow Contrastive Learning with Label Descriptions (FCLD) integrates label description representations to capture essential relational attributes, mitigating overfitting and reducing representation bias in few-shot settings. Comprehensive ablation studies and extensive experiments on two FCRE benchmarks demonstrate the effectiveness of each proposed component and superior performance of Minion over state-of-the-art methods, marking a significant contribution to the field of FCRE research.

6 Limitations

The current method is primarily applied to high-level relation extraction tasks where the entities are predefined. To create more practical and advanced FCRE systems, future research should focus on end-to-end relation extraction, which combines both entity recognition and relation extraction. This presents additional challenges, as it requires addressing overfitting and catastrophic forgetting across consecutive tasks.

While limited datasets present challenges in obtaining informative and robust representations, the use of data augmentation, which has been effectively integrated into previous FCRE methods (Qin and Joty, 2022; Ma et al., 2024; Tran et al., 2024), is an area our approach has not yet explored. Despite this limitation, Minion outperforms existing techniques that incorporate data augmentation for previously learned tasks in the replay buffer. This indicates that while our method successfully addresses the issue of analogous relations, there is still potential for improvement. We believe that integrating data augmentation could further enhance the performance of Minion, and we plan to explore this in future research.

Acknowledgments

This research has been supported by the NSF grant # 2239570. This research is also supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract 2022-22072200003. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government.

References

Nguyen Hoang Anh, Quyen Tran, Thanh Xuan Nguyen, Nguyen Thi Ngoc Diep, Linh Ngo Van, Thien Huu Nguyen, and Trung Le. 2025. Mutual-pairing data augmentation for fewshot continual relation extraction. In *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 4057–4075, Albuquerque, New Mexico. Association for Computational Linguistics.

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.

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.

Hien Dang, Tho Tran, Stanley Osher, Hung Tran-The, Nhat Ho, and Tan Nguyen. 2023. Neural collapse in deep linear networks: from balanced to imbalanced data. *arXiv preprint arXiv:2301.00437*.

Bao-Ngoc Dao, Quang Nguyen, Luyen Ngo Dinh, Minh Le, Nam Le, and Linh Ngo Van. 2025. Towards rehearsal-free continual relation extraction: Capturing within-task variance with adaptive prompting. *arXiv preprint arXiv:2505.13944*.

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, Dung Manh Nguyen, and Nghi D. Q. Bui. 2025. On the impacts of contexts on repository-level code generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1496–1524, Albuquerque, New Mexico. Association for Computational Linguistics.

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.

Jiale Han, Bo Cheng, and Wei Lu. 2021. Exploring task difficulty for few-shot relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2605–2616.

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.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

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*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2021. Supervised contrastive learning.

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.

Anh Duc Le, Nam Le Hai, Thanh Xuan Nguyen, Linh Ngo Van, Nguyen Thi Ngoc Diep, Sang Dinh, and Thien Huu Nguyen. 2025a. Enhancing discriminative representation in similar relation clusters for few-shot continual relation extraction. In *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, Albuquerque, New Mexico. Association for Computational Linguistics.

Minh Le, Tien Ngoc Luu, An Nguyen The, Thanh-Thien Le, Trang Nguyen, Tung Thanh Nguyen, Linh Ngo Van, and Thien Huu Nguyen. 2025b. 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.

Yang Liu, Jinpeng Hu, Xiang Wan, and Tsung-Hui Chang. 2022. A simple yet effective relation information guided approach for few-shot relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 757–763.

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.

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.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. 2017. No fuss distance metric learning using proxies.

Dung Nguyen, Le Nam, Anh Dau, Anh Nguyen, Khanh Nghiem, Jin Guo, and Nghi Bui. 2023a. The vault: A comprehensive multilingual dataset for advancing code understanding and generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4763–4788.

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

Toan Ngoc Nguyen, Nam Le Hai, Nguyen Doan Hieu, Dai An Nguyen, Linh Ngo Van, Thien Huu Nguyen, and Sang Dinh. 2025a. Improving Vietnamese-English cross-lingual retrieval for legal and general domains. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 142–153, Albuquerque, New Mexico. Association for Computational Linguistics.

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

Vardan Papyan, X. Y. Han, and David L. Donoho. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663.

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.

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

Yifan Song, Peiyi Wang, Weimin Xiong, Dawei Zhu, Tianyu Liu, Zhifang Sui, and Sujian Li. 2023. Infocl: Alleviating catastrophic forgetting in continual text classification from an information theoretic perspective.

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. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13771–13784, Miami, Florida, USA. Association for Computational Linguistics.

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.

Kaijia Yang, Nantao Zheng, Xinyu Dai, Liang He, Shujian Huang, and Jiajun Chen. 2020. Enhance prototypical network with text descriptions for few-shot relation classification. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 2273–2276.

Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. 2022. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network?

Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. 2023. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning.

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.

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; Le et al., 2024a). Several approaches have been explored and can be classified into three main categories: regularization/prior-based methods (Kirkpatrick et al., 2017; Van et al., 2022; Phan et al., 2022; Hai et al., 2024), 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 information extraction (Cui et al., 2021; Zhao et al., 2022; Le et al., 2024b; Dao et al., 2024, 2025).

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 Information Bottleneck and Representation Bias

According to Song et al. (2023), information bottleneck defines the objective of deep learning as balancing the trade-off between model generalization (compressing representations) and preserving information.

Given the input $\mathcal{X}$ and the class label $\mathcal{Y}$, the model is trained to learn the representation $\mathcal{Z} = f_{\mathcal{M}}(\mathcal{X})$. The purpose of fine-tuning model is to minimize the following Lagrangian:

$$I(\mathcal{X}; \mathcal{Z}) - \beta I(\mathcal{Z}; \mathcal{Y}), \tag{10}$$

where $I(\mathcal{X}; \mathcal{Z})$ is the mutual information (MI) between $\mathcal{X}$ and $\mathcal{Z}$, measuring the information retained in the representation $\mathcal{Z}$. $I(\mathcal{Z}; \mathcal{Y})$ quantifies the amount of information in $\mathcal{Z}$ that enables the identification of the label $\mathcal{Y}$. β is a trade-off hyperparameter. Using the information bottleneck principle, the model will learn *minimal sufficient representation* $\mathcal{Z}^*$ of $\mathcal{X}$ corresponding to $\mathcal{Y}$:

$$\begin{aligned} \mathcal{Z}^* &= \arg\min_{\mathcal{Z}} I(\mathcal{X}; \mathcal{Z}) \\ \text{s.t. } & I(\mathcal{Z}; \mathcal{Y}) = I(\mathcal{X}; \mathcal{Y}). \end{aligned} \tag{11}$$

The minimal sufficient representation is crucial for supervised learning, as it retains only the essential information about the input that is necessary to classify the label, thereby simplifying the classifier's role and enhancing generalization, while preserving all relevant label information.

In continual relation extraction setting, following information bottleneck, model learns the minimal sufficient representation $\boldsymbol{z}_{x_i}$ of the input during task $\mathcal{T}^i$ as:

$$\begin{aligned} \boldsymbol{z}_{x_i} &= \arg\min_{\boldsymbol{z}_{x_i}} I\left(\mathcal{F}(x); \boldsymbol{z}_{x_i}\right) \\ \text{s.t. } & I\left(\boldsymbol{z}_{x_i}; y_i\right) = I(\mathcal{F}(x); y_i), \end{aligned} \tag{12}$$

Nevertheless, this leads to a phenomenon known as ***representation bias***. To be more specific, in the i^{th} session, compressing crucial representation of local classes $r_j \in R_i$ may result in global insufficiency, which is expressed as:

$$I\left(\boldsymbol{z}_x; y\right) < I(\mathcal{F}(x); y). \tag{13}$$

As a result, the model tends to exhibit a bias toward classifying local relations within each session, which may limit its ability to capture sufficient information for distinguishing analogous classes in later sessions, leading to catastrophic forgetting.

C Experimental Details

C.1 Datasets

We conduct our experiments on two benchmark datasets:

- **TACRED** (Zhang et al., 2017) The TACRED dataset comprises 42 relations and 106,264 examples derived from Newswire and Web documents. Following the approach outlined by (Qin and Joty, 2022), we exclude instances annotated as "no_relation" and partition the remaining 41 relations into 8 distinct tasks. The first task, $\mathcal{T}^1$, consists of 6 relations, each containing 100 examples, while the subsequent tasks are configured as *5-way 5-shot* tasks, with each involving 5 relations.

- **FewRel** (Han et al., 2018) dataset, encompassing 100 relations and 70,000 examples, is adapted for our experiments following the setup proposed by Qin and Joty (2022). Specifically, 80 relations are organized into 8 tasks, each comprising 10 relations (*10-way*). While the first task $\mathcal{T}^1$ is designed with 100 examples per relation, the subsequent tasks are structured as few-shot learning scenarios, constrained to a *5-shot* setting.

C.2 Baselines

This section presents a concise summary of several state-of-the-art approaches in Few-Shot Continual Relation Extraction (FCRE), which are utilized as benchmark baselines in our evaluations.

- **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.

- **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** (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.

- **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.

- **CPL+MI+augment** (Anh et al., 2025) introduces a data augmentation strategy that enriches input by combining original and new information to create more complex texts. It also incorporates adversarial training and custom objective functions to enhance robustness and learning from diverse training signals.

- **SIRUS** (Le et al., 2025a) proposes a method to tackle the challenge of similar classes by representing relations through their descriptions and applying dynamic clustering to discover groups of semantically related relations.

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.

C.3 Large Language Model Embeddings

Although Large Language Models (LLMs) with billions of parameters excel at autoregressive text generation tasks (Dubey et al., 2024; Jiang et al., 2023; Hai et al., 2025; Nguyen et al., 2025a, 2023a), their generation-focused architecture often limits their effectiveness in text representation learning compared to discriminative encoder-based models like BERT. Large Language Model Embeddings (LLMEs) are introduced to transform decoder-only LLMs into text encoders, thereby enhancing their representation learning and embedding capabilities (BehnamGhader et al., 2024; Li et al., 2024; Lee et al., 2024). To this end, two key modifications are typically applied: (1) enabling bidirectional attention by removing the causal mask, and (2) replacing the next-token prediction task with contrastive learning or masked token prediction during training. As a result, these models can function similarly to encoder models such as BERT while offering more generalization and comprehension capabilities, since they can inherit the strengths of the extensive architecture and pretraining corpus of the original LLMs.

We investigate the use of Large Language Model Embeddings (LLMEs) in the FCRE scenario by utilizing the backbone model $\mathcal{M}$ with these models. Given that LLMs excel with instruction prompts and mean-pooling of token embeddings has been shown to yield optimal results in LLM2Vec (BehnamGhader et al., 2024), we construct an input x incorporating entities e_h and e_t as follows.

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

This instruction prompt enables LLMEs to capture the semantic context and classify relations between the entities. The latent embedding is then derived by mean-pooling the token representations. The training and inference processes are consistent across all backbone models.

C.4 Backbone Checkpoint

- For BERT-based models: We use BERT-base-uncased checkpoint[1] on Hugging Face.
- For LLM2Vec-based models: We employ three checkpoints on Huggingface:
  - *LLama3:* Meta-Llama-3-8B-Instruct-mntp-supervised [2](a variant of a Llama-3 8B model),
  - *Mistral:* LLM2Vec-Mistral-7B-Instruct-v2-mntp-unsup-simcse [3]
  - *LLama2:* LLM2Vec-Llama-2-7b-chat-hf-mntp-supervised [4] checkpoint on Hugging Face. (a variant of a Llama-2 7B model)

C.5 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 $\mathcal{T}^j$ 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 $\mathcal{T}^i$ after training the model on task $\mathcal{T}^j$.

Training Configuration: All BERT-based experiments were performed on an NVIDIA RTX 3090 GPU with 24GB of memory, while experiments using the LLME backbone were conducted on an NVIDIA A100 GPU with 80GB of VRAM. The experiments were carried out on 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, **0.1**, 0.15, 0.2, 0.25}
- λ_2: {0.5, **0.5**, 0.5, **0.5**, 0.5}
- λ_3: {**0.25**, **0.5**, 0.75, 1.0}

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

Additionally, Tables 4 and 5 provide the optimal values of hyperparameters for each model backbone.

Table 4: Hyperparameters setting for the BERT-backbone.

Hyperparameter	Value
Epochs	10
Learning rate	1×10^{-5}
α	0.25
θ (TACRED)	0.3
θ (FewRel)	0.1
Encoder output size	768
BERT input max length	256
λ_1	1.0
λ_2 (FewRel)	0.5
λ_2 (TACRED)	0.25
λ_3 (FewRel)	0.5
λ_3 (TACRED)	0.25
Soft prompt initialization	Random
Soft prompt phrase length	3
Soft prompt number of phrases	4

Table 5: Hyperparameters setting for LLMEs backbone.

Hyperparameter	Value
Encoder output size	4096
Epochs	10
Learning rate	1×10^{-5}, 1×10^{-4}
α	0.2
θ (FewRel)	0.5
θ (Tacred)	0.5
Lora alpha	16
Lora rank	8
Lora dropout	0.05
λ_1	1.0
λ_2	0.5
λ_3	0.5

D Additional Experimental Results

D.1 Efficiency of ProtoGOF in Discriminating Analogous Relations

Follow Song et al. (2023), we select the top most analogous labels in FewRel dataset, which are: P706, P57, P22, P123, P127, P25, P17, P551, P206, P58, P40, P35, P26, P131, P937. As demonstrated in Figure 2, ProtoGOF assists the model in distinguishing between similar classes more effectively. In particular, for the label "P123: Publisher" and the relation "P58: Screenwriter," it drastically reduces the number of incorrect predictions from 50 to just 11. Moreover, ProtoGof also helps to eliminate the confusion between similar relations as well, such as: similiar pair: relation "P35: Head of government" and "P937: Work location".

D.2 Additional Ablation Results

Effectiveness of each Proposed Components: Additionally, we conduct an ablation study to evaluate the impact of each component in Minion by systematically removing them from the overall objective function and framework. As shown in Table 6, the results confirm that each core component of Minion, including ProtoGOF and FCLD, is crucial for the model's performance. For a fair comparison, we also reproduce InfoCL (Song et al., 2023) with BERT as the baseline and use NC-FSCIL (Yang et al., 2023) with ETF-based prototypes (ProtoETF) instead of ProtoGOF. The results reveal that, while InfoCL achieves competitive performance in standard continual learning settings, it suffers significantly from catastrophic forgetting in few-shot tasks, particularly on TACRED (5-way-5-shot) with an accuracy of only 51.02%. Furthermore, despite the claim that ETF structure ensures "maximal separation," in imbalanced training settings like FCRE, deep learning models do not converge to ETF structures, but rather to GOFs. Consequently, ProtoETF shows a slight decline in performance compared to ProtoGOF. Moreover, we compare Minion with existing approaches leveraging $\mathcal{L}_{SCL}$(Khosla et al., 2021), such as RP-CRE (Cui et al., 2021) and CPL (Ma et al., 2024). Table 2 demonstrates that our method outperforms these models by a significant margin on both benchmark datasets.

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

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)*								
Minion	$94.58_{\pm 0.36}$	$\mathbf{86.96_{\pm 2.56}}$	$\mathbf{81.02_{\pm 1.33}}$	$\mathbf{77.66_{\pm 2.18}}$	$\mathbf{75.66_{\pm 1.96}}$	$\mathbf{73.05_{\pm 1.25}}$	$\mathbf{71.33_{\pm 1.16}}$	$\mathbf{69.61_{\pm 2.12}}$
w/o $ProtoGOF$	94.73	86.48	80.04	76.21	74.65	72.06	69.91	68.27
w/o $FCLD$	94.87	86.71	80.61	77.18	75.47	72.17	70.31	68.32
w $ProtoETF$	94.87	86.61	80.09	76.66	74.88	71.70	70.35	68.60
CPL (Ma et al., 2024)	94.87	85.14	78.80	75.10	72.57	69.57	66.85	64.50
InfoCL (Song et al., 2023)	94.56	86.17	79.61	75.80	73.67	70.22	68.15	65.72
RP-CRE (Cui et al., 2021)	$93.97_{\pm 0.64}$	$76.05_{\pm 2.36}$	$71.36_{\pm 2.83}$	$69.32_{\pm 3.98}$	$64.95_{\pm 3.09}$	$61.99_{\pm 2.09}$	$60.59_{\pm 1.87}$	$59.57_{\pm 1.13}$
TACRED *(5-way-5-shot)*								
Minion	$87.28_{\pm 0.23}$	$\mathbf{81.89_{\pm 1.35}}$	$\mathbf{75.72_{\pm 4.6}}$	$\mathbf{72.75_{\pm 3.94}}$	$\mathbf{66.68_{\pm 5.27}}$	$\mathbf{66.71_{\pm 6.6}}$	$\mathbf{62.81_{\pm 5.55}}$	$\mathbf{61.11_{\pm 2.58}}$
w/o $ProtoGOF$	87.28	83.16	75.98	74.67	68.43	67.66	62.67	59.66
w/o $FCLD$	87.12	83.16	75.98	72.73	67.24	65.97	61.70	59.49
w $ProtoETF$	86.96	83.03	76.48	73.02	66.65	65.87	61.91	60.28
CPL (Ma et al., 2024)	86.27	81.55	73.52	68.96	63.96	62.66	59.96	57.39
InfoCL (Song et al., 2023)	86.74	79.47	68.37	64.39	59.19	56.49	52.13	51.02
RP-CRE (Cui et al., 2021)	$87.32_{\pm 1.76}$	$74.90_{\pm 6.13}$	$67.88_{\pm 4.31}$	$60.02_{\pm 5.37}$	$53.26_{\pm 4.67}$	$50.72_{\pm 7.62}$	$46.21_{\pm 5.29}$	$44.48_{\pm 3.74}$

Table 6: Ablation study (%) of ProtoGOF and FCLD in our Minion with BERT baseline. We reproduced NC-FSCIL as ProtoETF and InfoCL in FCRE. The best results are in bold.

Hyper-parameters	$\mathcal{T}^1$	$\mathcal{T}^2$	$\mathcal{T}^3$	$\mathcal{T}^4$	$\mathcal{T}^5$	$\mathcal{T}^6$	$\mathcal{T}^7$	$\mathcal{T}^8$
Best setting	$94.58_{\pm 0.36}$	$86.96_{\pm 2.56}$	$81.02_{\pm 1.33}$	$77.66_{\pm 2.18}$	$75.66_{\pm 1.96}$	$73.05_{\pm 1.25}$	$71.33_{\pm 1.16}$	$69.61_{\pm 2.12}$
α: {0.1, 0.15, 0.2}								
0.1	$94.69_{\pm 0.32}$	$86.86_{\pm 2.45}$	$80.66_{\pm 2.04}$	$77.50_{\pm 2.76}$	$75.62_{\pm 2.95}$	$72.62_{\pm 2.01}$	$71.39_{\pm 1.19}$	$69.32_{\pm 0.80}$
0.15	$94.69_{\pm 0.32}$	$87.12_{\pm 2.23}$	$80.80_{\pm 1.75}$	$77.09_{\pm 2.57}$	$75.38_{\pm 2.60}$	$72.28_{\pm 1.73}$	$70.69_{\pm 1.14}$	$68.54_{\pm 0.64}$
0.2	$94.68_{\pm 0.32}$	$87.13_{\pm 2.44}$	$80.96_{\pm 1.54}$	$77.52_{\pm 2.17}$	$75.65_{\pm 2.43}$	$72.39_{\pm 1.89}$	$70.95_{\pm 1.49}$	$68.84_{\pm 1.18}$
λ_1: {0.1, 0.25, 0.5}								
0.1	$94.66_{\pm 0.32}$	$86.96_{\pm 2.50}$	$80.56_{\pm 1.75}$	$76.27_{\pm 2.60}$	$74.00_{\pm 2.67}$	$71.33_{\pm 1.72}$	$69.64_{\pm 1.44}$	$67.23_{\pm 0.92}$
0.25	$94.66_{\pm 0.32}$	$86.96_{\pm 2.28}$	$80.34_{\pm 1.98}$	$77.00_{\pm 2.83}$	$75.19_{\pm 2.90}$	$72.44_{\pm 1.96}$	$70.82_{\pm 1.27}$	$68.06_{\pm 0.74}$
0.5	$94.69_{\pm 0.32}$	$87.12_{\pm 2.52}$	$80.70_{\pm 1.74}$	$77.02_{\pm 2.33}$	$75.23_{\pm 2.77}$	$72.59_{\pm 2.09}$	$70.53_{\pm 1.22}$	$68.66_{\pm 0.65}$
λ_2: {0.25, 0.1, 1.0}								
0.25	$94.68_{\pm 0.32}$	$87.13_{\pm 2.24}$	$80.85_{\pm 1.91}$	$77.19_{\pm 2.31}$	$75.60_{\pm 2.91}$	$72.65_{\pm 2.06}$	$71.13_{\pm 0.91}$	$69.00_{\pm 0.96}$
0.1	$94.69_{\pm 0.32}$	$87.35_{\pm 2.24}$	$81.37_{\pm 2.08}$	$77.89_{\pm 2.56}$	$76.04_{\pm 2.72}$	$72.90_{\pm 1.89}$	$71.19_{\pm 1.28}$	$69.48_{\pm 0.80}$
1.0	$94.66_{\pm 0.32}$	$87.08_{\pm 2.19}$	$81.09_{\pm 1.88}$	$77.43_{\pm 2.48}$	$75.43_{\pm 2.70}$	$72.55_{\pm 2.41}$	$70.86_{\pm 1.16}$	$68.97_{\pm 0.77}$
λ_3: {0.25, 0.75, 1.0}								
0.25	$94.68_{\pm 0.32}$	$87.15_{\pm 2.30}$	$80.72_{\pm 1.52}$	$77.60_{\pm 2.08}$	$75.95_{\pm 2.51}$	$73.12_{\pm 1.98}$	$71.37_{\pm 1.08}$	$69.17_{\pm 0.70}$
0.75	$94.68_{\pm 0.32}$	$87.20_{\pm 2.56}$	$80.62_{\pm 2.22}$	$77.48_{\pm 1.61}$	$75.55_{\pm 2.04}$	$73.01_{\pm 1.93}$	$71.15_{\pm 1.25}$	$69.29_{\pm 0.71}$
1.0	$94.68_{\pm 0.32}$	$87.10_{\pm 2.46}$	$80.92_{\pm 2.00}$	$77.13_{\pm 2.60}$	$75.43_{\pm 2.62}$	$72.49_{\pm 1.62}$	$70.83_{\pm 1.10}$	$69.06_{\pm 0.84}$

Table 7: Performance of different hyper-parameter settings across tasks. Each cell shows the mean accuracy and standard deviation.

Hyperparameter sensitivity: We conducted additional experiments by varying the values of key hyperparameters, including α, λ_1, λ_2, and λ_3. We find out that the denotation for α is missing in Equation 8. To clarify, this parameter represents the weight for $\mathcal{L}_{\text{SCL}}$, and we have included it in our updated revision. Specifically, to reduce the exponential number of possible configurations, while tuning one parameter at a time we fix the remaining parameters to their optimal settings ($\alpha = \mathbf{0.25}$, $\lambda_1 = \mathbf{1.0}$, $\lambda_2 = \mathbf{0.5}$, $\lambda_3 = \mathbf{0.5}$). The table below provides the results for the FewRel dataset under different parameter settings. The results indicate that performance varies with a low standard deviation across different hyperparameter configurations, demonstrating that our method is not overly sensitive to these hyperparameters. Notably, the most sensitive parameter we observed is λ_1, which corresponds to $\mathcal{L}_{fs}$. This indicates that careful tuning of the loss between the input and label description from the fast and slow encoders is crucial for achieving rich representations. However, the variance in performance is not significantly high, and the other parameters exhibit robustness. This suggests that the proposed approach generalizes effectively to new datasets or tasks without requiring extensive hyperparameter tuning.

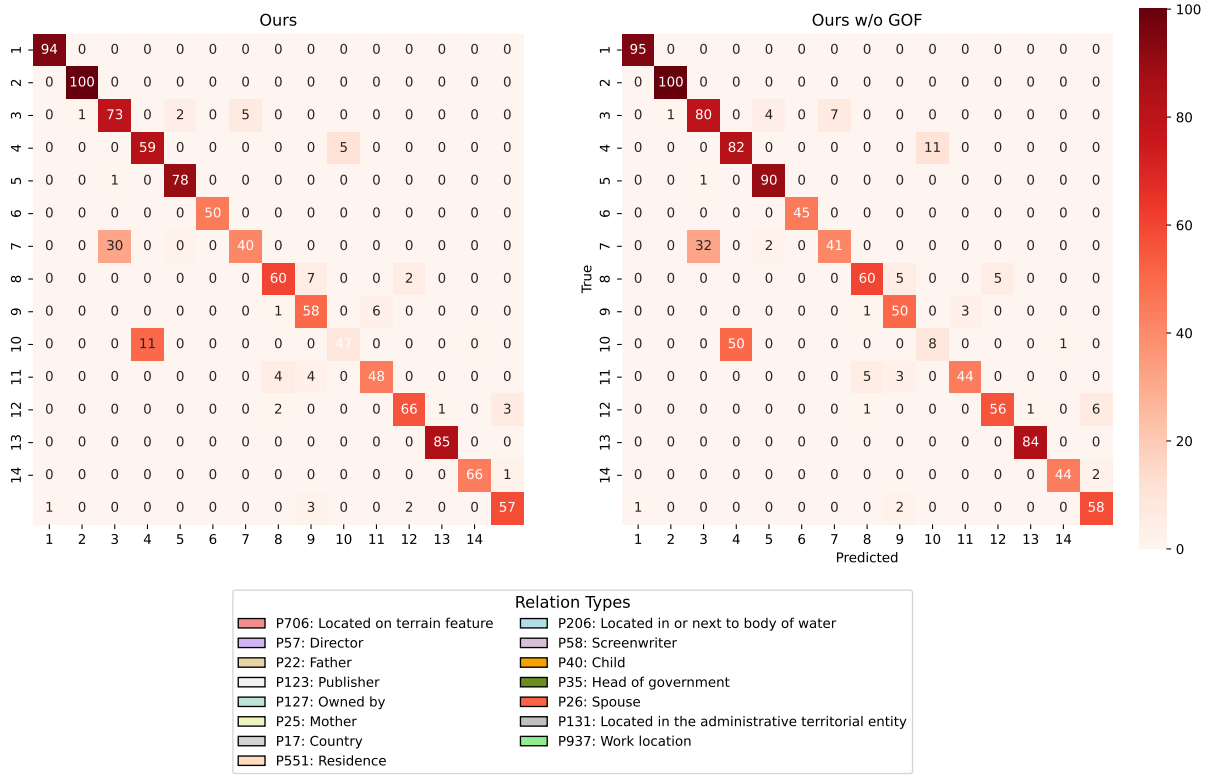


Figure 2: Confusion matrix between Minion and Minion w/o ProtoGOF for 15 similar relations after training 8 tasks in the FewRel dataset.