# Dynamic-prototype Contrastive Fine-tuning for Continual Few-shot Relation Extraction with Unseen Relation Detection

**Simiao Zhao[1], Zhen Tan[1*], Ning Pang[1] , Weidong Xiao[1], Xiang Zhao[2]**
[1]National Key Laboratory of Information Systems Engineering,
National University of Defense Technology, China
[2]Laboratory for Big Data and Decision,
National University of Defense Technology, China
{simiao_zhao2001, tanzhen08a, pangning14, xiangzhao}@nudt.edu.cn,
wilsonshaw@vip.sina.com

## Abstract

Continual Few-shot Relation Extraction (CFRE) aims to continually learn new relations from limited labeled data while preserving knowledge about previously learned relations. Facing the inherent issue of catastrophic forgetting, previous approaches predominantly rely on memory replay strategies. However, they often overlook task interference in continual learning and the varying memory requirements for different relations. To address these shortcomings, we propose a novel framework, DPC-FT, which features: 1) *a lightweight relation encoder* for each task to mitigate negative knowledge transfer across tasks; 2) *a dynamic prototype module* to allocate less memory for easier relations and more memory for harder relations. Additionally, we introduce the None-Of-The-Above (NOTA) detection in CFRE and propose a threshold criterion to identify relations that have never been learned. Extensive experiments demonstrate the effectiveness and efficiency of our method in CFRE, making our approach more practical and comprehensive for real-world scenarios.

## 1 Introduction

Relation Extraction (RE) aims to identify the semantic relation between two annotated entities in a text (Wang et al., 2023). Conventional RE methods (Zeng et al., 2014; Zhou et al., 2016; Zhang et al., 2018) typically rely on a fixed set of predefined relations and fixed training on a static dataset. However, this approach limits the ability to handle emergent relations outside the pre-defined set. A naive solution is to retrain the model with both historical and new relations, which results in increased computational and storage demands.

In response to the emergent relations in real applications, (Wang et al., 2019) propose the concept of Continual Relation Extraction (CRE), which attempts to continuously integrate newly introduced relations without degrading performance on previous ones. Considering the shortage of labeled data for relations, a more challenging task, Continual Few-shot Relation Extraction (CFRE), is proposed (Qin and Joty, 2022) to acquire relational knowledge only from a handful of labeled samples. As a typical continual learning process, CRE suffers the inherent catastrophic forgetting problem (Rebuffi et al., 2017), i.e., forgetting earlier relational knowledge. This is because the RE model requires to be trained on a sequence of tasks, where the relation distribution of each task progressively changes.

To overcome the forgetting issue, recent literature predominantly rely on memory-based methods and has established a two-stage training paradigm. Specifically, when a new task emerges, the RE model is first adapted to the new relations and then fine-tuned using a fixed number of memory samples for the current relations along with memory samples of historic relations (Han et al., 2020; Cui et al., 2021).

However, these memory-based CRE and CFRE methods assume that each task in the sequence is independent and the learning difficulty of each relation within a task remains consistent. In practice, these assumptions are often violated. On one hand, differences in relation distributions may cause negative knowledge transfer from past tasks to the current one. On the other hand, the intrinsic imbalance in learned relations means that allocating uniform memory size to all relations impedes the revisitation of those poorly remembered.

To address the two overlooked issues above, we propose a dynamic-prototype contrastive fine-tuning method, namely DPC-FT. Specifically, we use Low-Rank Adaptation (LoRA) (Hu et al., 2021) to fine-tune lightweight parameters per task and select the appropriate encoder for the test text. At the same time, we designed a dynamic prototype module to allocate memory based on the semantic complexity of each relation, where we assign less

[*]Corresponding author.

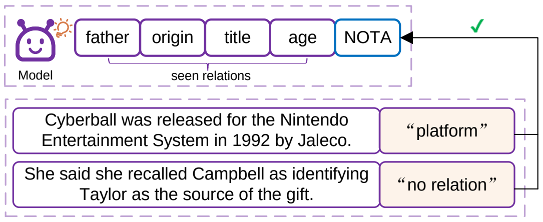
father
origin
title
age
NOTA
Model
seen relations
✓
Cyberball was released for the Nintendo Entertainment System in 1992 by Jaleco.
"platform"
She said she recalled Campbell as identifying Taylor as the source of the gift.
"no relation"

Figure 1: Examples of NOTA relation.

memory to easy relations and more memory to hard ones. The method not only aligns with the brain's natural ability to process different batches of knowledge in separate regions (Luppi et al., 2022) but also enabling to sufficiently revisit poorly learned relations.

Moreover, we also find that in real scenarios, it is quite common to encounter unstructured data that does not correspond to any of the learned relations, i.e., None-Of-The-Above (NOTA) relation. As shown in Fig. 1, for instance, we set up two sentences to illustrate the concept of NOTA data. The sentence "*Cyberball was released for the Nintendo Entertainment System in 1992 by Jaleco.*" represents an unseen relation for the model, while the latter one contains no relational semantics at all. Although NOTA is commonly addressed as an additional class in traditional RE tasks, detecting NOTA can be challenging in continual learning due to the limited memory space, which cannot fully encompass the complex semantic space of NOTA relations. Our study introduces the NOTA detection task into CRE for the first time and proposes a threshold criterion to address it. Specifically, during testing, we comprehensively scored the test texts and effectively filtered out the NOTA data.

**Contributions.** In this paper, we make the following contributions:

- We propose a model with lightweight LoRA to disentangle tasks for targeted encoding, and designed a dynamic prototype module to allocate the number of prototypes based on the semantic complexity of each relation.
- We pioneer the NOTA setting in the CRE field, proposing a threshold criterion that combines both angle and distance, providing a more comprehensive comparison of the similarity.
- Extensive experiments on FewRel and TACRED verify our models excellent performance on learned relations and strong results in identifying NOTA relations.

## 2 Related Work

Currently, memory-based methods have demonstrated higher effectiveness in CRE by storing representative data from previous tasks. However, recent CRE models, such as CML (Wu et al., 2021), RP-CRE (Cui et al., 2021), CRECL (Hu et al., 2022), CRL (Zhao et al., 2022), and KIP (Zhang et al., 2022), primarily focused on improving memory replay and activation, little investigation has been conducted on model training or memory sample selection. When the number of new tasks scales up, the approaches of refining only the memory replay stage tend to cause bias due to the high similarity and excessive number of memory samples, making it difficult to distinguish among them.

Due to the data sparsity inherent in few-shot scenarios and the challenge of obtaining high-quality data for each relation, CFRE tasks often suffer from overfitting and difficulties in semantic extraction. (Qin and Joty, 2022) first proposed applying few-shot settings in CRE tasks. SCKD (Wang et al., 2023) and ConPL (Chen et al., 2023) introduced methods demonstrating the feasibility of knowledge distillation and augmentation. However, they did not address filtering unseen relations. CPL (Ma et al., 2024) used ChatGPT for memory augmentation, but this approach heavily relies on ChatGPT and still involves storing many samples without refined management. PLE (Li et al., 2022) was proposed to tackle continual few-shot learning by training a specific prefix parameter for each task, effectively disentangling tasks and reducing mutual interference. It provides a new approach for continual few-shot learning tasks.

Parameter-Efficient Fine-Tuning (PEFT) techniques aim to adapt models to specific tasks by minimizing modifications to the pre-trained models. (Houlsby et al., 2019) pioneered PEFT research by designing an Adapter structure. Subsequently, methods such as Prefix Tuning (Li and Liang, 2021), LoRA (Hu et al., 2021), and BitFit (Zaken et al., 2021) were introduced, reducing overfitting risk while maintaining model generalization capabilities, and minimizing computational costs and memory requirements, making them particularly suitable for CFRE tasks.

Based on previous memory-based methods, our work leverages task disentanglement and PEFT techniques to address CFRE tasks, and designs a dynamic prototype method considering the semantic complexity differences across relations.

## 3 Methodology

In this section, we first formalize the task and then details the implementation of our proposed model.

### 3.1 Task Formalization

In the setting of N-way K-shot, the model needs to continually learn a series of tasks $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$. Each task $\mathcal{T}_k$ includes its own train, test and validate datasets, $\mathcal{D}_k^{\text{train}}$, $\mathcal{D}_k^{\text{test}}$ and $\mathcal{D}_k^{\text{val}}$, along with corresponding relations $\mathcal{R}_k$. The dataset $\mathcal{D}_k^{\text{train}}$ consists of $N$ relations, with each relation containing $K$ instances, thus forming $\mathcal{D}_k^{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N\times K}$. Each sample $(x_i, y_i)$ comprises a sentence $x_i$ with a pair of entities $(e_h, e_t)$ and a relation label $y_i \in \mathcal{R}_k$. In our experiments, we set $K$ to 10. The test dataset $\mathcal{D}_k^{\text{test}}$ contains $N$ relations, with each relation having 20 texts. Additionally, $\mathcal{D}_k^{\text{val}}$ includes an extra set of *"no relation"* types to evaluate the model's filtering performance on NOTA samples.

Different from previous memory-based methods, we do not need a memory bank to save the raw memory samples of each task. Instead, we propose a dynamic prototype module to memorize prototype representations corresponding to the relations in $\mathcal{R}_k$. The prototype repository contains all learned relations since the start of training and is dynamically updated. The model's performance is assessed using the cumulative validation sets of all tasks observed up to the current one, denoted by $\mathcal{D}_k^{\text{val}} = \bigcup_{i=1}^{k} \mathcal{D}_i^{\text{val}}$. This comprehensive evaluation requires the model to accurately recognize and classify relations across all texts it has been trained on.

### 3.2 The Overall of the Model

For the current task $\mathcal{T}_k$, the model's training and testing process for the given few-shot labeled data includes the following steps:

**Model Training**: For each task $\mathcal{T}_k$, we employ LoRA (Hu et al., 2021) as the lightweight fine-tuning method with low-rank matrices, significantly reducing parameter adjustment costs. During training, we store each task's fine-tuned parameter state to create task-specific encoder.

**Dynamic Prototype Module**: For each relation, we obtain its prototype representation through dynamic aggregation. We define an initial radius $r_0$ and a fluctuation radius $r_c$. In the embedding space, highly concentrated relations are averaged to obtain one embedding as a typical memory prototype. Conversely, for less concentrated types, the cluster centers are dynamically updated based on their shifts to obtain a more comprehensive prototype representation.

**Relation Classification**: For test texts with NOTA and known relations, we use our threshold criterion for classification. This module calculates the similarity scores between the test texts and prototype embeddings, considering both angle and distance factors, to complete relationship classification and filter out NOTA texts.

Next, we will provide a detailed explanation of each stage.

### 3.3 Relation Extraction Model

**Encoder** Following (Zhao et al., 2023), we use an entity marker-based encoder $\mathcal{E}(\cdot)$ for learning representations. Given an instance $x$ concerning entity pair $(e_h, e_t)$, four entity markers are placed around $(e_h, e_t)$ to denote the start and end positions:

$$\{\cdots, \text{[E1]}, e_h, \text{[/E1]}, \cdots, \text{[E2]}, e_t, \text{[/E2]}, \cdots\}. \tag{1}$$

Then, we feed the token sequence into a base version of BERT model to get the hidden vectors of [E1] and [E2], denoted as $\mathbf{h}_1$ and $\mathbf{h}_2 \in \mathbb{R}^d$. Finally, we get the instance-level representation as:

$$\mathbf{h} = \mathsf{LayerNorm}(\mathbf{W}_1[\mathbf{h}_1; \mathbf{h}_2] + \mathbf{b}_1), \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{d\times 2d}$ and $\mathbf{b} \in \mathbb{R}^d$ are learnable parameters, and [;] is the concatenation operation.

**Classifier** After obtaining the output of Encoder $\mathcal{E}(\cdot)$, the classifier figures out the relation probability of $x$:

$$p(y|x) = \mathsf{softmax}(\mathbf{W}_2\mathbf{h} + \mathbf{b}_2), \tag{3}$$

where $\mathbf{W}_2 \in \mathbb{R}^{N\times d}$ and $\mathbf{b}_2 \in \mathbb{R}^N$ are learnable parameters, and $N$ is the number of relations in the current task.

**Parameter-efficient fine-tuning** To reduce the training overhead, we apply LoRA to the BERT encoder by freezing its original weight matrix $\mathbf{W} \in \mathbb{R}^{d\times k}$ and incorporating trainable low-rank matrices $\mathbf{P} \in \mathbb{R}^{d\times r}$ and $\mathbf{Q} \in \mathbb{R}^{r\times k}$ to update the self-attention layers. This allows efficient fine-tuning on the current task without training all parameters. The weight update mechanism is as follows:

$$\mathbf{W}' = \mathbf{W} + \alpha \cdot \mathbf{PQ}, \tag{4}$$

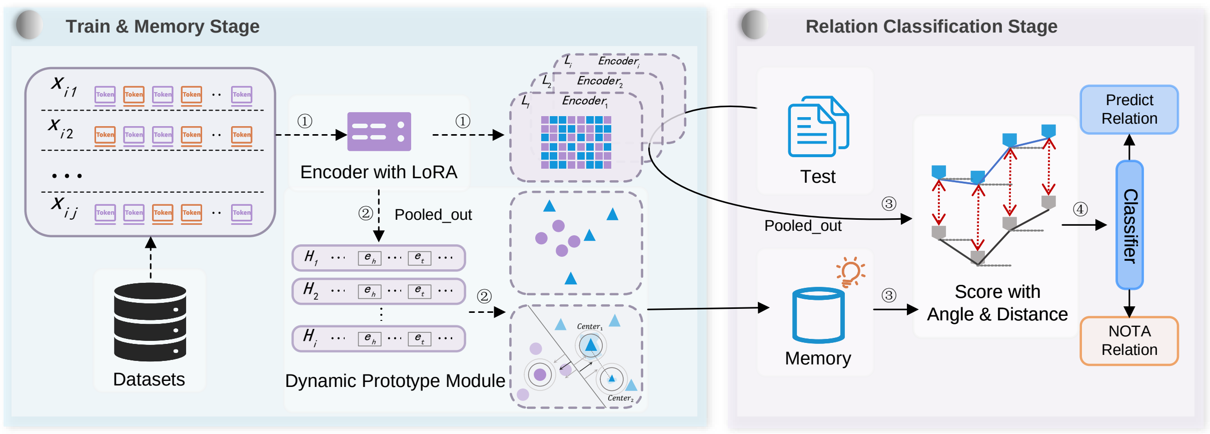

Figure 2: Our model's overall architecture is shown above. It includes: ① our lightweight multi-task encoding process; ② dynamic acquisition of relation memory prototypes through the encoder's embeddings, with the resulting prototypes stored in the memory pool; ③ the process where the test text and memory prototypes are input into the threshold criterion for similarity scoring; ④ filtering and classification to produce the final results.

where $\alpha$ acts as a scaling factor that modulates the impact of the update. We set the rank $r$ to 16, where $r \ll \min(d, k)$.

For each task $\mathcal{T}_k$ with $N \times K$ training instances, we individually learn and store a distinct set of $\mathbf{P}_k$ and $\mathbf{Q}_k$[1]. To optimize the parameters, we utilize the cross-entropy loss function as our primary loss function, defined as:

$$\mathcal{L}_1 = -\sum_{i=1}^{N\times K} \log p(y_i|x_i), \tag{5}$$

where $x_i$ is the $i$-th training instance in task $\mathcal{T}_k$, and $y_i \in \mathcal{R}_k$ is the ground truth of $x_i$.

## 3.4 Dynamic Prototype Module

**Dynamic prototype acquisition** The dynamic partitioning of embeddings is based on the Euclidean distance between a newly arrived embedding and existing prototype centers. For a target relation $r$ in task $\mathcal{T}_k$, we have a set of instance embeddings, denoted as $\{\mathbf{h}_1^r, \cdots, \mathbf{h}_K^r\}$. The dynamic process is illustrated in Algorithm 1. In Algorithm 1, $b$ is the number of embeddings in the current cluster, $\rho$ is the pre-defined cluster radius, and $\delta$ is an optional extension to the radius. This extension is utilized when $d_{\min}$ is slightly larger than $\rho$ but still within an acceptable range ($\rho + \delta$). After dynamic clustering, we can obtain a set of centroids for each relation $\mathbf{C}^r = \{\mathbf{c}_1, \cdots, \mathbf{c}_m\}$, which represents the relational prototypes.

[1]In the pre-trained Bert-large model, there are 343,882,836 parameters in total, while we fine-tuned only 8,693,802, making up just 2.53% of the total.

**Algorithm 1:** Dynamic prototype acquisition.

```
Input: $\{\mathbf{h}_1^r, \cdots, \mathbf{h}_K^r\}$
Initialize the centroid set as $\mathbf{C}^r = \emptyset$;
Initialize the number of clusters $m = 0$;
foreach $\mathbf{h}_i^r, i \in [1, \ldots, K]$ do
    if $i == 1$ then
        Obtain the $m$-th centroid $\mathbf{c}_m = \mathbf{h}_1^r$;
        $\mathbf{C}^r = \mathbf{C}^r \cup \{\mathbf{c}_m\}$ ;
    else
        foreach $\mathbf{c}_j \in \mathbf{C}^r$ do
            $d_{ij} = ||\mathbf{h}_i - \mathbf{c}_j||$;
        $d_{\min}, j^* = \min_j(d_{ij})$;
        if $d_{\min} \le \rho$ then
            $\mathbf{c}_{j^*} = \frac{1}{b+1}(b \cdot \mathbf{c}_{j^*} + \mathbf{h}_i^r)$;
        else if $\rho < d_{\min} \le \rho + \delta$ then
            $\mathbf{c}_{j^*} = \mathbf{c}_{j^*}, \rho = d_{\min}$;
        else
            $m = m + 1, \mathbf{c}_m = \mathbf{h}_i^r$;
            $\mathbf{C}^r = \mathbf{C}^r \cup \{\mathbf{c}_m\}$ ;
return The centroid set $\mathbf{C}^r$;
```

In this way, we dynamically adjust the embedding representations of training texts based on their spatial clustering, allowing for multi-center prototypes for complex semantic relations and single-center prototypes for simpler ones.

**Prototype contrastive learning** In this phase, we designed a contrastive loss function to improve clustering by better distinguishing different relations. In this phase, we employ a contrastive approach to design a similarity calculation loss that better evaluates the classification performance of new inputs. When a new training embedding arrives, we calculate its euclidean distance to all prototypes, classifying it by assigning the nearest prototype as the positive example and others as negative. If it forms a new prototype center, it becomes its own positive example. The loss is computed as follows:

$$\mathcal{L}_2 = 1 - \text{mean}(\sigma(\text{sim}_{\text{pos}}) - \sigma(\text{sim}_{\text{neg}})), \quad (6)$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\text{sim}_{\text{pos}}$ represents the cosine similarity of prototypes for same-class pairs, and $\text{sim}_{\text{neg}}$ for different-class pairs.

During training, we use a joint loss which is defined as:

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2, \quad (7)$$

where $\mathcal{L}_1$ focuses on optimizing relation prediction while $\mathcal{L}_2$ aims to enhance relation distinction within the embedding space.

### 3.5 Relation Inference

In the inference phase, we introduced a scoring mechanism to address bias in high-dimensional spaces, noting that relying solely on distance can cause inaccuracies. Our method balances cosine similarity and Euclidean distance, considering both distance and angle in the scoring function. Given a test sample $q$, we input it into $n$ lightweight relation encoders to generate its $n$ possible relation representations $\{\mathbf{q}_1, \cdots, \mathbf{q}_n\}$. For a target relation $r$ belonging to task $\mathcal{T}_k$, we retrieve $\mathbf{q}_k$ and compare it with prototypes $\{\mathbf{c}_1, \cdots, \mathbf{c}_m\}$. The scoring function is defined as:

$$s_{ki}^r = w_1 \cdot \frac{\mathbf{q}_k \cdot \mathbf{c}_i}{\|\mathbf{q}_k\|\|\mathbf{c}_i\|} + w_2 \cdot \frac{1}{1 + \|\mathbf{q}_k - \mathbf{c}_i\|}, \quad (8)$$

where $w_1$ and $w_2$ are set to a ratio of 3:7, representing the weights used to balance cosine similarity and distance similarity. The radio is chosen based on the optimal ratio determined through our experiments. Finally, the relation with the highest score for the test sample is selected as the predicted label:

$$r^* = \arg \max_{r \in \bigcup_{k=1}^{n} \mathcal{R}^k} \{s_{ki}^r\}. \quad (9)$$

Since we set the number of NOTA texts to be 10% of the validation set, we established our threshold at the lowest 12% of the similarity scores computed by the model from the validation set texts, allowing a slight margin. This accounts for cases where texts from the same class may exhibit significant semantic differences, causing classification challenges.

## 4 Experiments

### 4.1 Datasets

Our experiments were conducted on two widely used datasets.

**FewRel** It is a benchmark for relation extraction originally designed for few-shot learning, comprising 70,000 instances across 100 relations, and annotated by crowd workers to ensure reliability (Han et al., 2018). In our work, we follow prior research (Ma et al., 2024) and use a subset of 80 relations for the task.

**TACRED** It is a comprehensive benchmark for relation extraction, including 42 relations (a special 'no relation') and containing 106,264 samples from newswire and web documents (Zhang et al., 2017). Unlike FewRel, TACRED features an unbalanced distribution of relations, which better simulates real-world scenarios. Unlike previous methods, our approach uniquely integrates the 'no relation' category during the inference phase for None-Of-The-Above (NOTA) detection.

Previous works divided the relations in the dataset into 8 or 10 groups to simulate continual learning scenarios. Experimental results indicated that classification performance significantly declined as the number of tasks increased. Thus, we opted to randomly divide the datasets into 10 groups and select 10 texts per relation as training data, which is more challenging than pre-clustering semantically similar relations for each task (Wang et al., 2019; Han et al., 2020). Unlike earlier few-shot CRE methods that use a larger number of samples initially and impose strict few-shot constraints on later tasks (Qin and Joty, 2022; Wang et al., 2023), our strategy applies few-shot settings consistently across all tasks, thereby increasing the challenge.

### 4.2 Compared Models

We evaluate DPC-FT against seven recent CRE and CFRE models, including RP-CRE (Cui et al., 2021), CRL (Zhao et al., 2022), CRECL (Hu et al., 2022), ERDA (Qin and Joty, 2022), DP-CRE (Huang et al., 2024), SCKD (Wang et al., 2023), and ConPL (Chen et al., 2023). Since RP-CRE, CRL, CRECL, and DP-CRE do not address few-shot scenarios, and ERDA lacks a strict evaluation setup, we reproduced their results by running their source code. In terms of NOTA detection, we compared DPC-FT with LLMs in RQ3.

| Method | Task Index | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
| **8-way 10-shot of FewRel** | | | | | | | | | | |
| RP-CRE(Cui et al., 2021) | 95.39 | 82.36 | 76.23 | 74.34 | 73.36 | 73.69 | 72.38 | 71.34 | 69.52 | 66.97 |
| CRL(Zhao et al., 2022) | 95.12 | 88.17 | 85.37 | 83.52 | 80.85 | 78.48 | 72.46 | 69.05 | 68.23 | 65.86 |
| CRECL(Hu et al., 2022) | 96.12 | 89.82 | 86.23 | 83.97 | 78.03 | 75.82 | 71.48 | 71.32 | 68.84 | 68.76 |
| ERDA(Qin and Joty, 2022) | 93.25 | 79.82 | 72.56 | 69.88 | 62.23 | 58.62 | 55.22 | 50.51 | 49.34 | 46.98 |
| DP-CRE(Huang et al., 2024) | 85.67 | 83.22 | 80.96 | 77.31 | 76.54 | 71.93 | 71.64 | 70.82 | 70.14 | 69.39 |
| SCKD(Wang et al., 2023) | 97.79 | 93.16 | 92.72 | 90.36 | 88.64 | 86.79 | 81.68 | 78.16 | 73.22 | 70.68 |
| ConPL(Chen et al., 2023) | **98.01** | 94.23 | 92.67 | 91.22 | 89.94 | 88.16 | 86.43 | 84.68 | 82.14 | 80.97 |
| DPC-FT (Ours) | 97.89 | **95.81** | **93.08** | **92.37** | **92.08** | **88.69** | **88.63** | **87.32** | **87.52** | **85.96** |
| **4-way 10-shot of TACRED** | | | | | | | | | | |
| RP-CRE(Cui et al., 2021) | 92.95 | 88.48 | 82.25 | 78.23 | 76.65 | 72.43 | 66.34 | 60.81 | 60.28 | 58.84 |
| CRL(Zhao et al., 2022) | 90.12 | 88.89 | 85.17 | 82.68 | 80.43 | 76.48 | 72.25 | 69.43 | 65.05 | 56.51 |
| CRECL(Hu et al., 2022) | 88.68 | 82.85 | 80.43 | 78.25 | 73.88 | 64.79 | 62.55 | 60.24 | 58.54 | 54.23 |
| ERDA(Qin and Joty, 2022) | 84.42 | 76.85 | 72.23 | 64.95 | 56.28 | 50.67 | 43.12 | 39.95 | 36.34 | 34.46 |
| DP-CRE(Huang et al., 2024) | 84.68 | 83.95 | 80.23 | 77.82 | 76.43 | 76.02 | 75.74 | 74.23 | 72.67 | 70.72 |
| SCKD(Wang et al., 2023) | 93.82 | 89.35 | 84.07 | 83.68 | 78.53 | 73.17 | 71.23 | 68.52 | 64.55 | 63.14 |
| ConPL(Chen et al., 2023) | 97.82 | 94.54 | 90.32 | 89.95 | 85.23 | 83.76 | 80.17 | 79.21 | 78.98 | 76.43 |
| DPC-FT (Ours) | **98.36** | **96.97** | **95.35** | **90.57** | **86.79** | **87.18** | **81.77** | **82.32** | **81.86** | **80.08** |

Table 1: Result comparison on FewRel (8-way-10-shot) and TACRED (4-way-10-shot). The reported scores are the average of 5 training rounds.

### 4.3 Experimental Settings

Unlike other models that rely on extensive training data for the initial task followed by few-shot data for subsequent tasks, DPC-FT limits each task to 10 pieces of training instances to construct a 10-Shot setting. We adopted a strict evaluation method (Zhang et al., 2022), decreasing the semantic relevance of relationships within tasks and increasing task difficulty. Hyper-parameters were manually tuned. The basic parameter details used in the experiments can be found in Appendix A.3. To aid reproducibility, we will share the model's source code, detailed hyper-parameter settings, and processed samples.

### 4.4 Evaluation

We aim at answering the following research questions **RQs**:

- **RQ1**: What is the performance of DPC-FT compared to the prior baselines on CFRE benchmarks?
- **RQ2**:What is the effect of dynamic clustering method and the holistic scoring method?
- **RQ3**: How does the performance of DPC-FT compare to LLMs on CFRE and the filtering of NOTA samples?
- **RQ4**: What are the results of training time expenditure for our DPC-FT?

#### 4.4.1 RQ1: Overall Performance Comparison

The performance of our model and baselines is shown in Table 1, with scores averaged over 5 training rounds. The hyper-parameter configurations of baselines are the same as those reported in the original papers. The result of each task is the accuracy on the validation data of all observed relations. Based on the results, we find that:

(1) Our rigorous testing and sampling strategies have made CFRE tasks more challenging, leading to a notable decline in relation extraction performance for most models, particularly from the 7th to the 10th tasks. This decline is especially pronounced in the TACRED dataset, even with our strict sample limitations per relation.

(2) Compared to ConPL, the top-performing benchmark model, our model shows enhanced capabilities, particularly in reduced performance decline and steady accuracy gains. For example, our model's improvements range from 4.99% to 38.98% on the T10 metric for FewRel and from 3.65% to 45.62% on TACRED, demonstrating superior stability and performance.

(3) Experimental results reveal a significant drop in classification accuracy with new tasks, likely due to continuous parameter adjustments and strict control over training texts and memory samples.

(4) Most models show notably poor performance in T9 and T10, highlighting extensibility issues in few-shot CRE models. The influx of new tasks

| | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **FewRel** | | | | | | | | | | |
| DPC-FT | **97.89** | **95.81** | **93.08** | **92.37** | **92.08** | **88.69** | **88.63** | **87.32** | **87.52** | **85.96** |
| w/o dra. | 92.62 | 91.53 | 89.02 | 88.54 | 88.47 | 85.30 | 85.46 | 84.37 | 83.19 | 81.43 |
| w/o m-c. | 93.40 | 92.02 | 90.32 | 89.07 | 86.72 | 86.42 | 85.66 | 85.66 | 84.11 | 82.43 |
| w/o both | 89.49 | 88.15 | 86.70 | 85.80 | 83.67 | 83.59 | 83.04 | 83.03 | 81.90 | 80.47 |
| **TACRED** | | | | | | | | | | |
| DPC-FT | **98.36** | **96.97** | **95.35** | **90.57** | **86.79** | **87.18** | **81.77** | **82.32** | **81.86** | **80.08** |
| w/o dra. | 95.77 | 94.29 | 93.57 | 89.47 | 84.00 | 82.22 | 77.19 | 78.08 | 77.27 | 74.44 |
| w/o m-c. | 96.97 | 93.33 | 91.61 | 90.32 | 84.98 | 81.48 | 77.65 | 79.24 | 80.00 | 78.31 |
| w/o both | 95.24 | 91.67 | 89.29 | 86.90 | 83.33 | 80.95 | 78.57 | 75.00 | 72.62 | 70.24 |

Table 2: Ablation study on cluster modules.

| | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **FewRel** | | | | | | | | | | |
| DPC-FT | **97.89** | **95.81** | **93.08** | **92.37** | **92.08** | **88.69** | **88.63** | **87.32** | **87.52** | **85.96** |
| w/o cos-sim | 93.18 | 91.48 | 90.34 | 89.20 | 88.64 | 85.23 | 84.66 | 83.52 | 82.46 | 81.82 |
| w/o Euc-Dis. | 92.61 | 89.20 | 88.64 | 86.36 | 85.80 | 83.52 | 82.95 | 82.39 | 81.82 | 80.68 |
| **TACRED** | | | | | | | | | | |
| DPC-FT | **98.36** | **96.97** | **95.35** | **90.57** | **86.79** | **87.18** | **81.77** | **82.32** | **81.86** | **80.08** |
| w/o cos-sim | 94.05 | 92.86 | 90.48 | 85.71 | 83.33 | 82.14 | 78.57 | 77.38 | 76.19 | 75.00 |
| w/o Euc-Dis. | 91.67 | 90.48 | 88.10 | 85.71 | 80.95 | 79.76 | 77.38 | 76.19 | 75.00 | 73.81 |

Table 3: Ablation study on scoring modules.

impacts performance, likely from data confusion and discrimination challenges in a constrained embedding space. Our model's dynamic clustering approach addresses memory sample confusion to a certain extent, detailed further in RQ2.

(5) The results show that while most models perform well initially, our model maintains stability and excels in filtering NOTA samples as new tasks are added. These capabilities, detailed in RQ3, highlight our model's comprehensive performance.

#### 4.4.2 RQ2: The Effect of Dynamic Cluster and Holistic Scoring in DPC-FT

In this section, we present the ablation results for the dynamic prototype and holistic scoring modules, accompanied by a detailed analysis.

We first conduct the ablation study on our dynamic prototype module. In "w/o dra.", we disable the dynamic radius adjustment, forming a new cluster center whenever training samples deviate from the initial radius. In "w/o m-c.", we directly ignore samples outside the cluster center without creating additional centers, averaging only those within the dynamically adjusted radius to update the prototype. In "w/o both", we remove both modules.

From Table 2, we observe that: (1) Disabling any module reduces average accuracy, underscoring their importance; (2) Removing dynamic radius adjustment significantly lowers performance, and replacing both the dynamic radius and multi-center strategies with K-Means decreases accuracy by up to 9.84%, validating our proposed clustering approach; (3) When obtaining relation prototypes, fully considering semantic complexity and dynamically integrating representations is highly effective.

In the scoring module, our threshold criterion employs a scoring and selection method that combines Euclidean distance and cosine similarity in a 7:3 ratio, which has been proven to be the most effective through extensive testing. We also use only euclidean distance and cosine similarity for similarity calculations to test the method's validity.

Table 3 shows that using a single similarity calculation method lowers the classification accuracy, likely because both angle and distance are crucial for tensor similarity. Combining both provides a more effective similarity calculation for text semantic classification scenarios.

#### 4.4.3 RQ3: DPC-FT vs. General LLMs

Large language models like ChatGPT excel in relation extraction but their effectiveness in CFRE

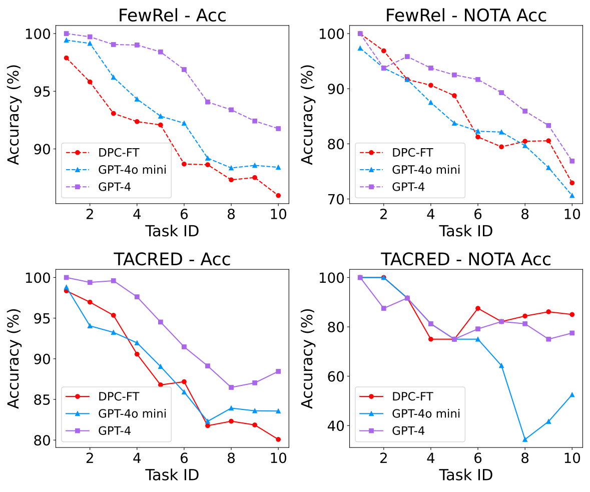

Figure 3: Classification accuracy of DPT-FT, GPT-4, and GPT-4o mini under experimental settings.

tasks is unexplored. This section compares DPC-FT with GPT-4 and GPT-4o mini under identical conditions, highlighting DPC-FT's strong performance and offering an objective analysis.

As shown in the Figure 3, the accuracy of the three models in relation prediction and NOTA data detection indicates that DPC-FT and GPT-4o mini exhibit comparable performance, with only a small gap compared to GPT-4. The experimental data can be found in Appendix A.3.

However, LLMs like ChatGPT face practical challenges such as security concerns and deployment difficulties, which limit their use and commercialization. GPT-4, with its 1.8 trillion parameters, requires high-end hardware, while GPT-4o mini, although smaller with 50 billion parameters, still surpasses the typical 7 billion parameters found in mobile edge models. In contrast, our DPC-FT utilizes BERT-large, which has only 340 million parameters and fine-tunes just 2.53% of parameters. Additionally, large models inherently benefit from their higher parameter counts and extensive pre-training data, which enhances their understanding of various relation types.

Our proposed DPC-FT achieves classification accuracy comparable to GPT-4o mini with significantly fewer parameters—adjusting only 4.9E-5 of GPT-4's parameter count. And it achieves near or even superior performance to GPT-4 in NOTA detection and surpasses GPT-4o mini. Beyond its impressive task performance and minimal training costs, DPC-FT also offers local security and flexibility, making it a more robust overall solution.

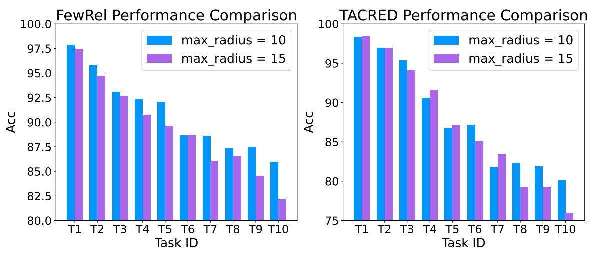

Figure 4: Comparison of maximum radius parameters in dynamic clustering.

### 4.4.4 RQ4: Analysis of Training Costs and Parameter Details

In this section, we provide an explanation and analysis of the details regarding training time and important hyper-parameters.

We tested various initial and maximum radius for dynamic prototype module, presenting results under six conditions. We chose the values of hyper-parameters with the closest performance for comparative visualization, specifically when the initial radius $r_o$ was 1.0 and the maximum radius $r_{max}$ was 10 or 15. Ultimately, we selected an initial radius of 1.0 and a maximum of 10.0, the experiment can be found in the appendix A.2.

Meanwhile, due to the minimal number of fine-tuning parameters, each training round takes only 4 seconds, significantly faster than other current models for the same task. Combined with the overall experimental results, our model delivers faster and more accurate performance.

## 5 Conclusion

In this paper, we introduce DPC-FT for the few-shot continual relation extraction task and, for the first time, propose an accuracy detection mechanism specifically for the None-Of-The-Above (NOTA) setting. To address the issue of catastrophic forgetting, we designed a method for selecting encoders and introduced a multi-center clustering approach with dynamically updated cluster radii. This ensures that the stored memory representations are more representative and comprehensive. Our experiments on FewRel and TACRED achieved state-of-the-art results, and extensive experiments further validated the effectiveness of each module. In future research, we plan to explore how the concepts of this model can be applied to solve more few-shot continual classification tasks, including few-shot multimodal relation extraction and better utilization of LLMs.

## 6 Limitations

This paper has two main limitations: (1) Compared to existing methods based on memory samples, DPC-FT may store more sets of encoder parameters. However, since we only fine-tuned a small portion of the parameters (2.53% of BERT), the difference in resource consumption is minimal compared to full fine-tuning of other models. (2) Although our dynamic clustering method reduces the number of memory samples, some relation types still retain more than three samples, causing a dispersion of meaning that needs addressing.

## Acknowledgments

We extend our gratitude to the anonymous reviewers, our meta-reviewer, and senior area chairs for their insightful comments and support throughout this work.This work was partially supported by National Key R&D Program of China (No. 2022YFB3103600), NSFC (Nos. U23A20296, 62272469, 72371245), and The Science and Technology Innovation Program of Hunan Province (No. 2023RC1007).

## References

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.

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.

Xu Han, Yi Dai, Tianyu Gao, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. Continual relation learning via episodic memory activation and reconsolidation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6440.

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. *arXiv preprint arXiv:1810.10147*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Chengwei Hu, Deqing Yang, Haoliang Jin, Zhen Chen, and Yanghua Xiao. 2022. Improving continual relation extraction through prototypical contrastive learning. *arXiv preprint arXiv:2210.04513*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Mengyi Huang, Meng Xiao, Ludi Wang, and Yi Du. 2024. Dp-cre: Continual relation extraction via decoupled contrastive learning and memory structure preservation. *arXiv preprint arXiv:2403.02718*.

Guodun Li, Yuchen Zhai, Qianglong Chen, Xing Gao, Ji Zhang, and Yin Zhang. 2022. Continual few-shot intent detection. In *Proceedings of the 29th international conference on computational linguistics*, pages 333–343.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Andrea I Luppi, Pedro AM Mediano, Fernando E Rosas, Negin Holland, Tim D Fryer, John T O'Brien, James B Rowe, David K Menon, Daniel Bor, and Emmanuel A Stamatakis. 2022. A synergistic core for human brain evolution and cognition. *Nature Neuroscience*, 25(6):771–782.

Shengkun Ma, Jiale Han, Yi Liang, and Bo Cheng. 2024. Making pre-trained language models better continual few-shot relation extractors. *arXiv preprint arXiv:2402.15713*.

Chengwei Qin and Shafiq Joty. 2022. Continual few-shot relation learning via embedding space regularization and data augmentation. *arXiv preprint arXiv:2203.02135*.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Sentence embedding alignment for lifelong relation extraction. *arXiv preprint arXiv:1903.02588*.

Xinyi Wang, Zitao Wang, and Wei Hu. 2023. Serial contrastive knowledge distillation for continual few-shot relation extraction. *arXiv preprint arXiv:2305.06616*.

Tongtong Wu, Xuekai Li, Yuan-Fang Li, Gholamreza Haffari, Guilin Qi, Yujin Zhu, and Guoqiang Xu. 2021. Curriculum-meta learning for order-robust continual relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10363–10369.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pages 2335–2344.

Han Zhang, Bin Liang, Min Yang, Hui Wang, and Ruifeng Xu. 2022. Prompt-based prototypical framework for continual relation extraction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2801–2813.

Runyan Zhang, Fanrong Meng, Yong Zhou, and Bing Liu. 2018. Relation classification via recurrent neural network with attention and tensor layers. *Big Data Mining and Analytics*, 1(3):234–244.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Conference on empirical methods in natural language processing*.

Kang Zhao, Hua Xu, Jiangong Yang, and Kai Gao. 2022. Consistent representation learning for continual relation extraction. *arXiv preprint arXiv:2203.02721*.

Wenzheng Zhao, Yuanning Cui, and Wei Hu. 2023. Improving continual relation extraction by distinguishing analogous semantics. *arXiv preprint arXiv:2305.06620*.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 207–212.

## A More Experimental Details and Results

### A.1 Experimental Parameters and Setup

We utilize a single NVIDIA GeForce RTX 4090 GPU with 24 GB memory on a 12th Gen Intel(R) Core(TM) i7-12700KF CPU @ 3.60GHz to run all experiments. For the hyperparameter search, we conduct a grid search to choose the appropriate values. Hyperparameters are illustrated in Table 4.

| Hyperparameters | Values |
|---|---|
| seed | 100 |
| batch_size | 40 |
| num_train_epochs | 100 |
| bert_learning_rate | 1e-5 |
| learning_rate | 1e-4 |
| BERT_hidden_size | 1024 |
| hidden_dropout_prob | 0.1 |
| Encoder_output_size | 1024 |
| max_seq_length | 128 |
| pre_seq_len | 80 |
| optimizer | AdamW |
| lr_lora_params | 5e-4 |
| lr_non_lora_params | 1e-4 |
| LoRA_r | 16 |
| LoRA_alpha | 32 |
| LoRA_target_modules | "query", "value" |
| LoRA_dropout | 0.1 |
| LoRA_bias | "none" |

Table 4: Hyperparameters setting.

### A.2 Dynamic Clustering Radius Selection Experiments

In our proposed dynamic clustering algorithm, we conducted detailed experiments on the initial radius $r_o$ and the maximum radius $r_{max}$ to select the most appropriate initial values. The experimental results are shown in Table 5.

Based on the experiments, we selected 1.0 as the initial radius and 10 as the maximum radius.

### A.3 Experiments on General LLMs Testing

We conducted CFRE experiments using the same dataset on both GPT and GPT-4o mini, with the results presented in Table 6, where we highlight the performance differences between DPC-FT, GPT-4o mini, and GPT-4. The results show that our model achieved excellent performance, particularly in the accuracy of NOTA sample extraction. Additionally, our model's relationship classification accuracy is comparable to that of GPT-4o mini and not far behind GPT-4. Notably, our model requires significantly fewer parameters than GPT-4o mini, meaning that we achieved strong relationship classification and outstanding NOTA sample classification with minimal resources.

| Dataset | radius | T1 | | T2 | | T3 | | T4 | | T5 | | T6 | | T7 | | T8 | | T9 | | T10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | max_radius | | | | | | | | | | | | | | | | | | | |
| | | 10 | 15 | 10 | 15 | 10 | 15 | 10 | 15 | 10 | 15 | 10 | 15 | 10 | 15 | 10 | 15 | 10 | 15 | 10 | 15 |
| FewRel | 1.5 | 97.27 | 96.80 | 93.88 | 94.54 | 92.67 | 91.84 | 91.17 | 90.58 | 89.48 | 89.58 | 88.33 | 88.31 | 86.03 | 84.07 | 83.83 | 83.86 | 82.16 | 84.07 | 81.67 | 80.74 |
| | 1.0 | 97.89 | 97.41 | 95.81 | 94.72 | 93.08 | 92.70 | 92.37 | 90.74 | 92.08 | 89.65 | 88.69 | 88.70 | 88.63 | 86.03 | 87.32 | 86.54 | 87.52 | 84.54 | 85.96 | 82.16 |
| | 0.5 | 97.27 | 96.72 | 93.58 | 90.90 | 91.31 | 91.79 | 90.50 | 90.50 | 89.72 | 89.48 | 87.47 | 87.05 | 84.14 | 83.75 | 82.18 | 83.86 | 81.04 | 82.18 | 80.11 | 79.51 |
| TACRED | 1.5 | 96.97 | 96.97 | 93.33 | 96.00 | 92.81 | 93.48 | 87.98 | 90.04 | 84.52 | 86.82 | 83.33 | 84.80 | 79.76 | 83.08 | 78.57 | 79.81 | 77.38 | 78.92 | 75.60 | 75.97 |
| | 1.0 | 98.36 | 98.41 | 96.97 | 96.94 | 95.35 | 94.07 | 90.57 | 91.61 | 86.79 | 87.10 | 87.18 | 85.05 | 81.77 | 83.39 | 82.32 | 79.21 | 81.86 | 79.21 | 80.08 | 75.98 |
| | 0.5 | 96.97 | 98.41 | 94.23 | 96.00 | 91.60 | 92.81 | 86.90 | 90.50 | 83.33 | 87.57 | 83.80 | 83.88 | 78.57 | 79.86 | 78.10 | 78.77 | 77.86 | 78.28 | 76.19 | 75.91 |

Table 5: The selection of clustering radius parameters.

| Dataset | Method | T1 | | T2 | | T3 | | T4 | | T5 | | T6 | | T7 | | T8 | | T9 | | T10 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **NOTA** | Acc | **NOTA** | Acc | **NOTA** | Acc | **NOTA** | Acc | **NOTA** | Acc | **NOTA** | Acc | **NOTA** | Acc | **NOTA** | Acc | **NOTA** | Acc | **NOTA** | Acc |
| FewRel | GPT-4 mini | 93.75 | 99.43 | 93.75 | 99.15 | 91.67 | 96.21 | 87.50 | 94.32 | 83.75 | 92.84 | 82.29 | 92.23 | 82.14 | 89.20 | 79.68 | 88.35 | 75.69 | 88.57 | 70.63 | 88.41 |
| | GPT-4 | 100 | 100 | 93.75 | 99.72 | 95.83 | 99.05 | 93.75 | 99.01 | 92.50 | 98.41 | 91.67 | 96.88 | 89.29 | 94.07 | 85.94 | 93.39 | 83.33 | 92.42 | 76.68 | 91.76 |
| | Ours | 100 | 97.89 | 96.88 | 95.81 | 91.67 | 93.08 | 90.63 | 92.37 | 88.75 | 92.08 | 81.25 | 88.69 | 79.46 | 88.63 | 80.47 | 87.32 | 80.56 | 87.52 | 72.92 | 85.96 |
| | Diff_4o mini | **+6.25** | -1.54 | **+3.13** | -3.34 | **0.00** | -3.13 | **+3.13** | -1.95 | **+5.00** | -0.76 | -1.04 | -3.54 | -2.68 | -0.57 | **+0.79** | -1.03 | **+4.87** | -1.05 | **+2.29** | -2.45 |
| | Diff_4 | **0.00** | -2.11 | **+3.13** | -3.91 | -4.16 | -5.97 | -3.12 | -6.64 | -3.75 | -6.33 | -10.42 | -8.19 | -9.83 | -5.44 | -5.47 | -6.07 | -2.77 | -4.90 | -3.96 | -5.80 |
| TACRED | GPT-4 mini | 100 | 98.81 | 100 | 94.05 | 91.67 | 93.25 | 81.25 | 91.96 | 75.00 | 89.05 | 75.00 | 85.91 | 64.29 | 82.31 | 34.38 | 83.93 | 41.67 | 83.60 | 52.50 | 83.57 |
| | GPT-4 | 100 | 100 | 87.5 | 99.40 | 91.67 | 99.60 | 81.25 | 97.62 | 75.00 | 94.52 | 79.17 | 91.47 | 82.14 | 89.12 | 81.25 | 86.48 | 75.00 | 87.04 | 77.50 | 88.45 |
| | Ours | 100 | 98.36 | 100 | 96.97 | 91.67 | 95.35 | 75.00 | 90.57 | 75.00 | 86.79 | 87.50 | 87.18 | 82.14 | 81.77 | 84.38 | 82.32 | 86.11 | 81.86 | 85.00 | 80.08 |
| | Diff_4o mini | **0.00** | -0.45 | **0.00** | **+2.92** | **0.00** | **+2.10** | -6.25 | -1.39 | **0.00** | -2.26 | **+12.50** | **+1.27** | **+17.85** | -0.54 | **+50.00** | -1.61 | **+44.44** | -1.74 | **+32.50** | -3.49 |
| | Diff_4 | **0.00** | -1.64 | **+12.50** | -2.43 | **0.00** | -4.25 | -6.25 | -7.05 | **0.00** | -7.73 | **+8.33** | -4.29 | **0.00** | -7.35 | **+3.13** | -4.16 | **+11.11** | -5.18 | **+7.50** | -8.37 |

Table 6: CRE experiment on general LLMs.