

# Actively Supervised Clustering for Open Relation Extraction

Jun Zhao<sup>1\*</sup>, Yongxin Zhang<sup>1\*</sup>, Qi Zhang<sup>1†</sup>, Tao Gui<sup>2†</sup>,  
Zhongyu Wei<sup>3</sup>, Minlong Peng<sup>4</sup>, Mingming Sun<sup>4</sup>

<sup>1</sup>School of Computer Science, Fudan University

<sup>2</sup>Institute of Modern Languages and Linguistics, Fudan University

<sup>3</sup>School of Data Science, Fudan University

<sup>4</sup>Cognitive Computing Lab Baidu Research

{zhaoj19, yongxinzhang20, qz, tgui, zywei}@fudan.edu.cn

## Abstract

Current clustering-based Open Relation Extraction (OpenRE) methods usually adopt a two-stage pipeline. The first stage simultaneously learns relation representations and assignments. The second stage manually labels several instances and thus names the relation for each cluster. However, unsupervised objectives struggle to optimize the model to derive accurate clustering assignments, and the number of clusters has to be supplied in advance. In this paper, we present a novel setting, named actively supervised clustering for OpenRE. Our insight lies in that clustering learning and relation labeling can be alternately performed, providing the necessary guidance for clustering without a significant increase in human effort. The key to the setting is selecting which instances to label. Instead of using classical active labeling strategies designed for fixed known classes, we propose a new strategy, which is applicable to dynamically discover clusters of unknown relations. Experimental results show that our method is able to discover almost all relational clusters in the data and improve the SOTA methods by 10.3% and 5.2%, on two datasets respectively.

## 1 Introduction

Relation extraction (RE) aims to detect and extract the potential relation between the given entity pair in unstructured text. The extracted relation facts play a vital role in many downstream applications, such as knowledge base population (Ji and Grishman, 2011), search engine (Schlichtkrull et al., 2018), and question answering (Yu et al., 2017). To deal with the emerging unknown relational types in the real world, Open Relation Extraction (OpenRE) has been widely studied.

The clustering-based unsupervised relation discovery is a classical paradigm for OpenRE (Yao

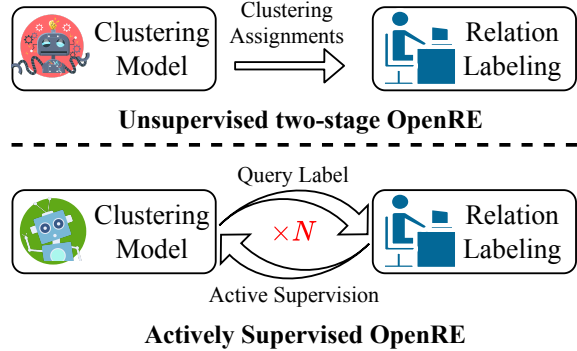


Figure 1: Compared with the existing unsupervised two-stage methods, our method can provide explicit supervision for clustering by alternately performing clustering learning and relation labeling. Note that the human effort of the two settings is comparable.

et al., 2011; Marcheggiani and Titov, 2016; Elsahar et al., 2017). It can discover potential relations, by grouping several instances into relational clusters, and then manually labeling a few instances to name the relation of each cluster. Recently, Hu et al. (2020) introduced a deep clustering framework (Caron et al., 2018) into OpenRE. They iteratively cluster the relation representations that are produced by large pretrained models and use the cluster assignments as pseudo-labels to refine the representations. Unfortunately, the above unsupervised methods struggle to learn good enough representations, and the cluster assignments are error-prone. When multiple relations are mixed in a cluster, it becomes difficult to name the cluster. Hence, instead of regarding OpenRE as a totally unsupervised task, researchers leverage the labeled data of predefined relations to provide explicit supervision signals for clustering learning (Wu et al., 2019; Zhao et al., 2021), and achieve superior results.

Different from the above two-stage methods, in this work, we present a new setting named actively supervised clustering for OpenRE (ASCORE). As

\*Equal Contributions.

†Corresponding authors.

shown in fig. 1, our insight lies in that clustering learning (i.e, deep clustering) and relation labeling can be alternately performed. In an iteration, a small number of key instances are selected for labeling. The unknown relations expressed by these instances are correspondingly discovered. More importantly, these labeled instances can provide explicit supervisory signals for clustering learning. The improved relation representations form a better cluster structure, which in turn is able to benefit the discovery of the neglected relations. Since potential relations are dynamically discovered in iterations, the number of clusters does not need to be provided in advance.

Along with this setting, we design an active labeling strategy tailored for clustering. First, all instances are encoded to points in the representation space, where the clustering is performed. The goal of the strategy is to select the most informative points for labeling. Intuitively, two points that are far from each other in representation space usually express different relations. To discover as many relations as possible, we introduce a distance regularization to the strategy, so that diversified relation discovery can be facilitated. To prevent over-fitting caused by training with limited active labeled instances, all the selected key points are required to be the points of maximum local density. By doing so, a large number of high-quality pseudo-labels can be obtained, by assigning active labels to unlabeled data in a small neighborhood. To mitigate the error propagation issue, different loss functions are assigned to active labels and pseudo-labels with different reliability for clustering learning. Experimental results show that (1) the actively supervised method improves the SOTA two-stage methods by a large margin without a significant increase in human effort. (2) the proposed active strategy can discover more relational clusters, compared with the classical active strategy.

To summarize, the main contributions of this work are as follows: (1) We present a new setting named actively supervised clustering for OpenRE, providing the necessary guidance for clustering without a significant increase in human effort. (2) Design of a new active labeling strategy tailored for clustering, that can effectively discover potential relational clusters in unlabeled data. (3) This method improves the SOTA two-stage methods by 10.3% and 5.2% on two well-known datasets,

respectively.

## 2 Related Work

**Clustering-based OpenRE:** The clustering-based paradigm considers relation discovery as a two-stage pipeline, which clusters relational data first, and then manually labels relational semantics for each cluster. Conventional methods cluster instances by human-defined linguistic features (Yao et al., 2011; Marcheggiani and Titov, 2016; Elsahar et al., 2017), such as entity words/type, dependency paths, trigger words, context POS tags. Recently, many studies have shown that pretrained models learn diversified linguistic knowledge (Jawahar et al., 2019; Clark et al., 2019; Goldberg, 2019; Zhao et al., 2022). Hu et al. (2020) leverage the self-supervised signals provided by the pretrained model to iteratively learn relation representations and optimize clustering. Due to the lack of strong supervision, it is difficult for the above methods to produce satisfactory clustering results. Although some works (Wu et al., 2019; Zhao et al., 2021) try to use the labeled data of predefined relations to complete the missing supervision, the semantic gap between predefined and open relations leads to negative clustering bias, especially when these relations come from different domains (Zhao et al., 2021). By performing clustering learning and relation labeling alternately, our actively supervised method can provide strong supervision and improve the two-stage methods by a large margin. *In the main results (sec. 5), we achieve this improvement at the cost of only two active labels for each relation on average. For two-stage methods, relation labeling for each cluster requires at least one (usually more) instance to be manually observed.* Therefore, there is no significant increase in human effort.

**Active Learning:** Active learning is a research field with high relevance to the proposed methods. In the research field, a classical method is uncertainty-based sampling (Roth and Small, 2006; Wang and Shang, 2014a; Tong and Koller, 2001). The uncertainty can be defined based on the posterior probability of a predicted class or the distances to the decision boundaries. In the context of deep learning, MC Dropout (Gal et al., 2017) is an effective way for uncertainty estimation, but the computationally inefficient limits its application in large-scale datasets. Recently, representative sampling is attracting lots of attention (Sener and

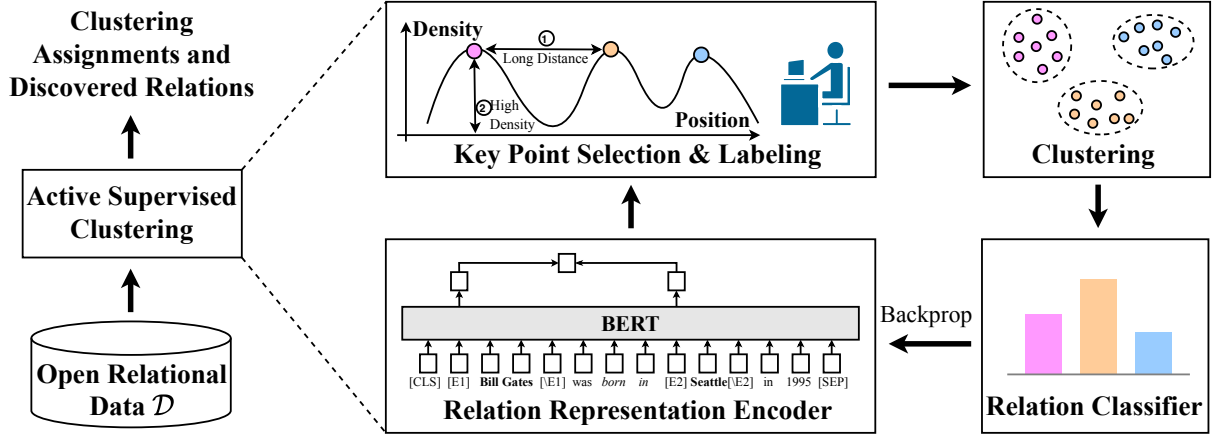


Figure 2: Overview of the training pipeline for our actively supervised clustering setting. In each iteration, a few key points are selected for relation labeling. The rest instances are clustered to the nearest key points. Some highly reliable cluster assignments are used as pseudo-labels for relation representation learning.

Savarese, 2018; Ash et al., 2019a), which selects data points that represent the distribution of an unlabeled pool. Unlike the classical labeling strategies designed for fixed classes, the proposed strategy is encouraged to discover new relational clusters while improving the clustering of relations that have been discovered.

### 3 Approach

In this work, we present a new setting named actively supervised clustering for OpenRE (ASCORE), which fuses the isolated two-stage pipeline to guide clustering learning. Fig. 2 illustrates the training pipeline. The OpenRE problem addressed in this work is formally stated as follows. Given as input an open relational dataset  $\mathcal{D} = \{x_i | i = 1, \dots, N\}$ , the goal is to discover and label potential relations  $\mathcal{R} = \{r_i | i = 1, \dots, K\}$  in the open data and cluster the corresponding instances. Note that the number of relations  $K$  in  $\mathcal{D}$  is *unknown*.

#### 3.1 Overview

The ASCORE is based on deep clustering (Caron et al., 2018), which is a common practice for OpenRE (Hu et al., 2020; Zhao et al., 2021), that iteratively clusters the representations of input instances, and uses the cluster assignments as pseudo-labels to learn the relation representations. We introduce explicit supervision to deep clustering by alternately performing clustering learning and relation labeling. The actively labeled points can serve as a basis to facilitate accurate pseudo-label estimation and improve representation learning. The improved relational representation in turn

benefits relation labeling to discover more relations. As illustrated in Figure 2, the training pipeline of ASCORE consists of the following steps:

**Encoding Step:** This step aims to obtain the relation representation  $h_i$  of each input instance  $x_i \in \mathcal{D}$ , laying the groundwork for clustering and relation discover. First, the contextual information of  $x_i$  is encoded to the entity pair representation  $h_i^{ent}$ , using a pretrained BERT (Devlin et al., 2018) encoder. To avoid data sparsity and low efficiency of clustering in high-dimensional space, an autoencoder is used to transform  $h_i^{ent}$  into a low-dimensional clustering-friendly representation  $h_i$ .

**Labeling Step:** This step aims to discover potential relations in open dataset  $\mathcal{D}$  and guide clustering. At the  $s^{\text{th}}$  iteration, a set  $\mathcal{D}_s^* \in \mathcal{D}$  is actively labeled, including  $B$  key points. These key points are required to be local maximum in density, so a large number of high-quality pseudo labels can be obtained by assigning active labels to unlabeled data in a small neighborhood. Instead of focusing only on the improving clustering of the discovered relations, all key points in  $\mathcal{D}^* = \mathcal{D}_1^* \cup \dots \cup \mathcal{D}_s^*$  are required to be far away from each other to facilitate the discovery of new relations.

**Learning Step:** This step aims to learn clustering relational data, using an actively labeled  $\mathcal{D}^*$ . Specifically, each unlabeled point  $x_i \in \mathcal{D}$  is clustered to the nearest key point  $x_j^* \in \mathcal{D}^*$  and the pseudo label is  $\hat{y}_i = y_j^*$ . The reliability of  $\hat{y}_i$  increases as the distance in representation space between  $x_i$  and  $x_j^*$  decreases. Cross-entropy loss (resp. divergence-based contrastive loss) is

used for pseudo labels with high (resp. moderate) reliability, to optimize relation representations and thus to improve clustering. With the help of active supervision, separated subclusters expressing the same relation approach each other, while mixed subclusters expressing different relations are separated. Existing unsupervised methods are inherently difficult to handle such errors.

The above three steps are performed iteratively to gradually improve the model performance. In the following sections, we will elaborate on the model structure, labeling strategy, and training methods involved in the above three steps.

### 3.2 Relation Representation Encoder

Given a relation instance  $x_i = \{w_1, w_2, \dots, w_n\}$ , in which four reserved word pieces  $[E1], [\backslash E1], [E2], [\backslash E2]$  are used to mark the beginning and end of each entity, the relation representation encoder  $f$  aims to encode the contextual relational information of instance  $x_i$  into a fixed-length representation  $\mathbf{h}_i^{ent} = f(x_i) \in \mathbb{R}^d$ . The encoder  $f$  is implemented as BERT. Specifically:

$$\mathbf{h}_1, \dots, \mathbf{h}_n = \text{BERT}(w_1, \dots, w_n) \quad (1)$$

$$\mathbf{h}^{ent} = \langle \mathbf{h}_{[E1]} | \mathbf{h}_{[E2]} \rangle. \quad (2)$$

Following Soares et al. (2019), the fixed-length representations are obtained, by concatenating the hidden states of marker  $[E1], [E2]$  (i.e.,  $\mathbf{h}_{[E1]}, \mathbf{h}_{[E2]}$ ) and  $\langle \cdot | \cdot \rangle$  is the concatenation operator.

However, clustering data in high-dimensional space is time-consuming and the data sparsity leads to sub-optimal clustering results. Therefore, an autoencoder is trained by reconstruction loss  $\mathcal{L}_{rec}$  and the encoder part is retained to transform high-dimensional  $\mathbf{h}^{ent}$  to a low-dimensional clustering-friendly relation representation  $\mathbf{h}$ .

### 3.3 Key Point Selection Module

In this section, the proposed key point selection method will be explained, including the labeling strategy and the conditions for stopping annotation. **Labeling strategy.** The labeling strategy is based on the following criteria. **First**, the selected key points are the local maximum in density. Generally, labels do not drastically change within a small neighborhood, and therefore, the first criteria enable to find a lot of unlabeled data within a small neighborhood of each key point, and accurately estimate their pseudo-labels. To find these local maximum, it is calculated the euclidean distance

between the relation representations  $\{\mathbf{h}_i\}_{i=1,2,\dots,N}$  obtained in encoding step, and the distance matrix  $\mathbf{D} \in \mathbb{R}^{N \times N}$  is constructed as follows:

$$D_{ij} = \|\mathbf{h}_i - \mathbf{h}_j\|_2^2, \quad (3)$$

where  $D_{ij}$  is the distance between two relational instance  $x_i$  and  $x_j$ . The potential computational cost to process large-scale datasets can be solved by sampling a small subset. Based on distance matrix  $\mathbf{D}$ , a density  $\rho_i$  is further defined for each relation instance  $x_i$ . A larger  $\rho_i$  indicates a larger number of instances around  $x_i$ :

$$\rho_i = \sum_{j=1}^N \text{sign}(D_c - D_{ij}), \quad (4)$$

where  $\text{sign}()$  is the sign function and  $D_c$  is a threshold.

To avoid the problem that all the labeled points are concentrated in several high-density areas and missing most long-tail relations, the **second** criteria is to keep these key points away from each other in clustering space. Specifically, a sparsity index  $\xi_i$  is defined for each instance  $x_i \in \mathcal{D}$ .

$$\xi_i = \begin{cases} \min_{j, \rho_j > \rho_i} D_{ij}, & \rho_i < \rho_{max} \\ \max_j D_{ij} & \rho_i = \rho_{max} \end{cases} \quad (5)$$

Intuitively, a larger  $\xi_i$  indicates that the instance  $x_i$  is a local maximum of density in a larger radius. Based on the density  $\rho_i$  and sparsity index  $\xi_i$  of each instance, the labeling strategy can be formally stated as follows. *In each iteration, choose  $B$  points with the highest density and their distance from each other is greater than  $\xi_c$ .*

$$\mathcal{D}_s^* = \text{TopB}_\rho \{x_i | \xi_i > \xi_c, x_i \in \mathcal{D}\} \quad (6)$$

To effectively support iterative labeling and maintain the diversity of key points, in  $s^{\text{th}}$  iteration, each new key point  $x_i$  should be as far away from the existing key point in  $\mathcal{D}^* = \mathcal{D}_1^* \cup \dots \cup \mathcal{D}_{s-1}^*$  as possible. Therefore, for each instance  $x_i$ , the sparsity index are modified as follows:

$$d = \min_{x_j \in \mathcal{D}^*} \|\mathbf{h}_i - \mathbf{h}_j\|_2^2 \quad (7)$$

$$\xi_i = \min(\xi_i, d), \quad (8)$$

After the  $s^{\text{th}}$  iterations, the result is the new active labeled set  $\mathcal{D}^* = \mathcal{D}^* \cup \mathcal{D}_s^*$ .

**Conditions for stopping annotation.** Too few queries will lead to missing some relations, while



too many queries will lead to unnecessary costs. Here we give a simple strategy to determine when to stop labeling. (1) First, users can determine the maximum number of actively labeled instances,  $N^*$ , based on their annotation budget. (2) Since our labeling strategy takes into account the diversity of key points, new relations are constantly discovered during the initial phase of iteration. When no new relations are discovered in two or more consecutive iteration steps (it means that most relations have been found), the labeling should be stopped.

### 3.4 Training Methods

**Pseudo Label Estimation.** Given the active labeled set  $\mathcal{D}^*$ , each of the rest unlabeled points  $x_i \in \mathcal{D}$  is clustered to the nearest key point  $x_j^* \in \mathcal{D}^*$  and  $\hat{y}_i$  is estimated as  $y_j^*$ . Intuitively, the accuracy of the pseudo label decreases with the increase of the distance between  $x_i$  and  $x_j^*$ . The reliability  $r$  of pseudo labels is defined as follows:

$$r_i = \|\mathbf{h}_i - \mathbf{h}_j^*\|_2^{-1}, \quad (9)$$

where  $\mathbf{h}_i$  and  $\mathbf{h}_j^*$  denote the representation of  $x_i$  and  $x_j^*$ , respectively.  $\|\cdot\|_2^{-1}$  denotes reciprocal of  $L_2$  norm.

**Model Optimization.** Given the pseudo label  $\hat{y}_i$  and its reliability  $r_i$  for each unlabeled data  $x_i \in \mathcal{D}$ , the relation representation is refined to improve clustering in the next iteration. Specifically, we first filter out a highly reliable subset  $\mathcal{D}_h = \{(x_i, \hat{y}_i) | r_i > r_h\}$  and use a softmax classifier to convert entity pair representation  $\mathbf{h}_i^{ent}$  into the probability distribution on discovered relations (denoted as  $\mathcal{P}_i$ ). The model is optimized with cross entropy loss for fast convergence:

$$\mathcal{L}_{ce} = \text{CrossEntropy}(\hat{y}_i, \mathcal{P}_i). \quad (10)$$

Note that the number of instances in  $\mathcal{D}_h$  is small. To avoid that the model only learns simple features, the threshold is broaden, and a moderately reliable subset  $\mathcal{D}_m = \{(x_i, \hat{y}_i) | r_i > r_m\}$  containing more instances is built. To mitigate the negative impact of noise in  $\mathcal{D}_m$ , a binary contrastive loss is further introduced:

$$\mathcal{L}_{bce} = \mathcal{L}_{\hat{y}_i=\hat{y}_j} + \mathcal{L}_{\hat{y}_i \neq \hat{y}_j} \quad (11)$$

$$\mathcal{L}_{\hat{y}_i=\hat{y}_j} = \mathcal{D}_{kl}(\mathcal{P}_i^* || \mathcal{P}_j) + \mathcal{D}_{kl}(\mathcal{P}_i || \mathcal{P}_j^*) \quad (12)$$

$$\mathcal{L}_{\hat{y}_i \neq \hat{y}_j} = H_\sigma(\mathcal{D}_{kl}(\mathcal{P}_i^* || \mathcal{P}_j)) + H_\sigma(\mathcal{D}_{kl}(\mathcal{P}_i || \mathcal{P}_j^*)) \quad (13)$$

$$H_\sigma(x) = \max(0, \sigma - x), \quad (14)$$

---

### Algorithm 1: ASCORE

---

**Input:** A open dataset  $\mathcal{D} = \{x_i\}_{i=1}^N$

- 1 **repeat**
- 2     Perform the encoding step (sec. 3.2).  
       Get relation representation  $\{\mathbf{h}_i\}_{i=1}^N$  for instances in  $\mathcal{D}$ ;
- 3     **if** *The conditions for stopping annotation are not met* **then**
- 4         Perform the labeling step (sec. 3.3).  
        Get the  $s^{\text{th}}$  active labeled  $\mathcal{D}_s$ ;  
         $\mathcal{D}^* = \mathcal{D}^* \cup \mathcal{D}_s$ ;
- 5     **end**
- 6     Perform the learning step (sec. 3.4).  
       Estimate pseudo labels and reliability scores. Use the corresponding loss for representation learning.
- 7 **until** *convergence*;
- 8 **Return** Discovered relation set  $\mathcal{R}$  and the cluster assignment  $\hat{y}_i \in \mathcal{R}$  of each instance  $x_i \in \mathcal{D}$ .

---

where  $\sigma$  is a hyperparameter and  $\mathcal{P}^*$  denotes that  $\mathcal{P}$  is assumed to be a constant for asymmetry.  $\mathcal{D}_{kl}$  denotes KL divergence. The probability distribution  $\mathcal{P}$  will be pulled closer or farther depending on whether the labels of the sample pairs are the same. In each iteration, if the annotator finds new relations, the parameters of the softmax classifier are reinitialized to deal with the new relations. Alg.1 shows an algorithm flow that is able to clearly summarize the proposed method.

## 4 Experimental Setup

### 4.1 Datasets

Experiments are conducted on two standard and one constructed dataset. Note that the compared baselines follow different settings. As will be described in sec. 4.2, RSN and RoCORE leverage labeled data of predefined relations, while RW-HAC and SelfORE follow unsupervised setting. To fairly compare all methods in a uniform setting, the first half of the relations in each dataset are held out as predefined relations. Specifically, in TACRED, 21 relations are held out, while in FewRel and FewRel-LT, the number is 40.

**TACRED** (Zhang et al., 2017). TACRED is a large-scale manually annotated RE dataset, covering 41 relations. Following Wu et al. (2019); Hu et al. (2020); Zhao et al. (2021), the instances labeled as `no_relation` are removed and the rest are used

for training and evaluation.

**FewRel** (Han et al., 2018). FewRel is a manually annotated dataset that contains 80 types of relations, each of which has 700 instances. However, in real-world OpenRE scenarios, unseen relations in unlabeled data usually follow a long-tailed distribution. To eliminate this inconsistency and accurately evaluate model performance in real-world scenarios, we construct a long-tail FewRel dataset as follows.

**FewRel-LT**. The FewRel-LongTail dataset. We number the 40 unseen relations in FewRel from 0 to 39, and calculate the number of samples according to  $y = \frac{700}{0.5 * i * d + 1}$ . The number of samples in the predefined subset remains unchanged.

## 4.2 Compared Methods

To evaluate the effectiveness of the actively supervised clustering, the following SOTA two-stage OpenRE methods are used for comparison.

**HAC with Re-weighted Word Embeddings (RW-HAC)** (Elsahar et al., 2017). A clustering-based OpenRE method. The model constructs relational feature based on entity types and the weighted sum of pretrained word embeddings.

**Relational Siamese Network (RSN)** (Wu et al., 2019). This method learns similarity metrics of relations from labeled data of pre-defined relations and then transfers the relational knowledge to identify novel relations in unlabeled data.

**Self-supervised Feature Learning for OpenRE (SelfORE)** (Hu et al., 2020). SelfORE exploits weak, self-supervised signals in pretrained language model for adaptive clustering on contextualized relational features.

**A Relation-oriented Clustering Method (RoCORE)** (Zhao et al., 2021). RoCORE leverages the labeled data of predefined relations to learn a clustering-friendly representation, which is used for new relations discovery.

To show the superiority of the proposed labeling strategy, the actively supervised clustering is combined with the following classical active learning strategies for comparison. Specifically, **RANDOM**, **CONFIDENCE** (Wang and Shang, 2014b), **MARGIN** (Roth and Small, 2006), **ENTROPY** (Wang and Shang, 2014a) and **GRADIENT** (Ash et al., 2019b) are included. We provide a brief introduction to these methods in appendix A.1.

## 4.3 Implementation Details

Following Hu et al. (2020) and Wu et al. (2019), 20% of the data in each dataset are held out for validation and hyperparameter selection. We use the Adam (Kingma and Ba, 2014) as the optimizer, with a learning rate of  $1e - 4$  and batch size of 100 for all datasets. The threshold  $D_c$  is given by the value of an element ranked top 40% in D from large to small. For each iteration, we label  $B = 20$  samples.  $\xi$  is set to the value when the number of candidates is  $1.2B$ . Some important hyperparameters  $r_h$  and  $r_m$  are analyzed in sec. 6.3. For a fair comparison, all active strategies select the same number of key points for labeling. Specifically, 40, 80 and 80 key points are labeled on the three datasets TACRED, FewRel, and FewRel-LT respectively. All experiments are conducted with Pytorch 1.7.0, using an NVIDIA GeForce RTX 3090 with 24GB memory.

## 5 Main Results

Table 1 shows the model performances on three datasets. In this section, the experiment focuses on the following two questions.

### 5.1 Does inaccurate estimation of the number of relations have an impact on clustering?

One drawback of the most existing two-stage OpenRE methods is that the number of clusters  $K$  has to be given in advance, which is impractical in real applications. When  $K$  is underestimated (from  $\text{💡}$  to  $\text{💡}$ ), the clustering performance of the SOTA unsupervised method, SelfORE, on the three datasets decreases by an average of 7.13%, while the same metric regarding RoCORE, the SOTA supervised method, is 18.10%. Furthermore, it is observed an extremely unbalanced precision-recall value in the  $B^3$  metric (much lower precision and higher recall), which indicates that the model tends to mix multiple relations in the same cluster. Obviously, such clustering results will have a negative impact on the relation labeling. In other words, it is difficult to determine which relation this cluster corresponds to. When  $K$  is overestimated (due to space limitation, please look at table 4 for results of overestimation), the same relation tends to be clustered into multiple subclusters. Repeated labeling of these subclusters brings a significant increase in human effort. In contrast, the ASCORE dynamically discovers relational clusters through active iteration, breaking the impractical
























Dataset	Method	Setting	$B^3$			V-measure			ARI	Classification		
			Prec.	Rec.	$F_1$	Hom.	Comp.	$F_1$		Prec.	Rec.	$F_1$
TACRED	RW-HAC  (Elsahar et al., 2017)	U	0.317	0.668	0.430	0.443	0.668	0.532	0.291	0.244	0.246	0.171
	SelfORE  (Hu et al., 2020)	U	0.339	0.759	0.469	0.468	0.809	0.593	0.412	0.121	0.244	0.160
	SelfORE  (Hu et al., 2020)	U	0.517	0.441	0.476	0.631	0.600	0.615	0.434	0.343	0.396	0.360
	RSN  (Wu et al., 2019)	P	0.312	0.807	0.451	0.445	0.768	0.563	0.354	0.149	0.118	0.225
	RoCORE  (Zhao et al., 2021)	P	0.361	0.930	0.520	0.525	0.903	0.664	0.447	0.116	0.247	0.152
	RoCORE  (Zhao et al., 2021)	P	0.696	0.685	0.690	0.786	0.786	0.787	0.640	0.547	0.594	0.563
	<b>Ours</b> 	A	0.742	0.821	<b>0.780</b>	0.807	0.856	<b>0.831</b>	<b>0.781</b>	0.698	0.715	<b>0.699</b>
FewRel	RW-HAC  (Elsahar et al., 2017)	U	0.175	0.367	0.237	0.357	0.463	0.403	0.108	0.251	0.264	0.216
	SelfORE  (Hu et al., 2020)	U	0.365	0.710	0.482	0.620	0.800	0.699	0.368	0.282	0.442	0.327
	SelfORE  (Hu et al., 2020)	U	0.527	0.552	0.539	0.728	0.736	0.732	0.517	0.604	0.632	0.600
	RSN  (Wu et al., 2019)	P	0.174	0.640	0.274	0.389	0.659	0.489	0.173	0.112	0.239	0.134
	RoCORE  (Zhao et al., 2021)	P	0.446	0.901	0.600	0.701	0.922	0.797	0.448	0.320	0.476	0.358
	RoCORE  (Zhao et al., 2021)	P	0.806	0.843	<b>0.824</b>	0.883	0.896	0.889	<b>0.807</b>	0.827	0.868	<b>0.837</b>
	<b>Ours</b> 	A	0.799	0.841	0.820	0.888	0.901	<b>0.894</b>	0.801	0.832	0.862	<b>0.838</b>
FewRel-LT	RW-HAC  (Elsahar et al., 2017)	U	0.255	0.322	0.285	0.379	0.421	0.399	0.145	0.190	0.176	0.160
	SelfORE  (Hu et al., 2020)	U	0.266	0.633	0.374	0.466	0.676	0.552	0.290	0.079	0.154	0.099
	SelfORE  (Hu et al., 2020)	U	0.563	0.456	0.504	0.717	0.661	0.687	0.377	0.439	0.526	0.462
	RSN  (Wu et al., 2019)	P	0.211	0.500	0.297	0.350	0.510	0.415	0.193	0.098	0.173	0.117
	RoCORE  (Zhao et al., 2021)	P	0.382	0.858	0.528	0.571	0.873	0.691	0.400	0.123	0.217	0.151
	RoCORE  (Zhao et al., 2021)	P	0.662	0.717	0.689	0.800	0.801	0.800	0.581	0.507	0.538	0.517
	<b>Ours</b> 	A	0.650	0.845	<b>0.735</b>	0.790	0.885	<b>0.835</b>	<b>0.676</b>	0.530	0.609	<b>0.550</b>

Table 1: Main results on three relation extraction datasets.  and  represent that the number of relations is known and unknown, respectively (please look at appendix A.2 for more details). U, P and A respectively indicate Unsupervised setting, supervised by Predefined relation setting and Actively supervised setting. The proposed method outperforms the SOTA method and does not need to specify the number of clusters in advance.

assumption that  $K$  is known in advance.

## 5.2 Is the actively supervised setting better than the two-stage setting?

The two settings are compared, from the following two perspectives.

In terms of clustering performance, the actively labeled data can provide valuable supervision signals for clustering learning. Compared with RoCORE, a strong baseline supervised by predefined relations, the proposed method improves the four metrics by an average of 10.3% and 5.2% on long-tail TACRED and FewRel-LT, respectively. Long-tail relation distribution is very common in real world. On the uniform FewRel dataset, the ASCORE achieves comparable results. It is worth noting that the number of clusters  $K$  has to be supplied for RoCORE. When  $K$  is unknown, the improvement will be even enlarged.

Regarding labeling costs, both settings are comparable. Note that in the main results, only two instances for each relation are labeled on average. For the two-stage methods, in order to label the relational semantics of a cluster, the annotator has to observe at least one sample. Obviously, the ASCORE does not lead to a significant increase in human efforts.

Dataset	RAND.	CONF.	MARG.	ENTRO.	GRAD.	OURS
TACRED	15	13	18	16	17	18
FewRel	40	34	40	37	40	40
FewRel-LT	27	19	25	26	30	35
ALL	82(↓ 5.7%)	66(↓ 24.1%)	83(↓ 4.6%)	79(↓ 9.2%)	87(0%)	<b>92(↑ 5.8%)</b>

Table 2: The number of relations discovered by the labeling strategies. (GRAD. as a reference point.)

## 6 Analysis and Discussions

### 6.1 Analysis on Labeling Strategy

Main results (sec. 5) have shown the advantages of the proposed actively supervised clustering setting over the two-stage setting. *However, some serious readers may still think the comparison across settings is unfair.* To further reduce readers' concerns and show the effectiveness of our labeling strategy, we combine the actively supervised clustering settings with the various active labeling strategies and compare them in terms of relation discovery and clustering performance. Note that the number of key points selected by each strategy is the same (two points per relation on average). The results are shown in tab.2 and tab. 3. It can be seen from tab. 2 that the proposed labeling strategy finds the most relations. Different from the classical strategies that focus only on improving the recognition of relations that have been discovered, the proposed strategy appropriately explores new relations by distance regularization,

Dataset	Method	$B^3$	V-measure	ARI	Classification
TACRED	RANDOM	0.737	0.800	0.662	0.464
	CONFIDENCE	0.671	0.752	0.598	0.408
	MARGIN	0.709	0.787	0.628	0.524
	ENTROPY	0.702	0.790	0.633	0.502
	GRADIENT	0.767	0.831	0.725	0.670
	<b>Ours</b>	<b>0.780</b>	<b>0.851</b>	<b>0.781</b>	<b>0.699</b>
FewRel	RANDOM	0.808	0.882	0.787	0.813
	CONFIDENCE	0.752	0.851	0.660	0.701
	MARGIN	0.817	0.881	0.796	0.831
	ENTROPY	0.781	0.819	0.779	0.743
	GRADIENT	0.814	0.884	0.790	0.827
	<b>Ours</b>	<b>0.820</b>	<b>0.894</b>	<b>0.801</b>	<b>0.838</b>
FewRel-LT	RANDOM	0.716	0.814	0.670	0.486
	CONFIDENCE	0.346	0.514	0.217	0.336
	MARGIN	0.696	0.806	0.647	0.481
	ENTROPY	0.721	0.824	0.664	0.481
	GRADIENT	0.719	0.797	0.649	0.498
	<b>Ours</b>	<b>0.735</b>	<b>0.835</b>	<b>0.676</b>	<b>0.550</b>

Table 3: Comparison results of labeling strategies on three datasets.

which is particularly beneficial for long-tail relation discovery in real applications. Additionally, tab. 3 shows that this strategy is also the best in terms of clustering performance. Benefitting from assigning reasonable loss functions to pseudo labels with different reliability, more pseudo labels can be used for learning without significantly increasing the risk of over-fitting noise.

## 6.2 The Effect of Numbers of Actively Labeled Instances

In order to more comprehensively evaluate the labeling strategies, experiments are conducted to compare these labeling strategies by varying the number of actively labeled instances,  $N^*$ . Figure 3 shows the effect of  $N^*$ . Surprisingly, it is found that the random strategy is a very competitive baseline that beats most of the classical labeling strategies given different  $N^*$ . This suggests that classical labeling strategies may be better at tasks with known and fixed categories. Although the proposed strategy consistently outperforms all baselines, it is obvious that it has not been fully optimized. It is the authors’ belief that it is sufficient to serve as a reasonable baseline for the actively supervised clustering setting and proceeding some useful guidance for future research in this field. Additionally, with the increase of  $N^*$ , the performance of the model is improved, but the growth rate is gradually slow. It means that the cost performance of human effort gradually decreases. Therefore, for users with limited budgets, it is also a good choice to discover the primary relations through only a few queries.

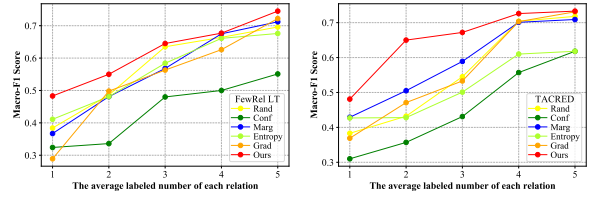


Figure 3: Performance with different active labeled instances.

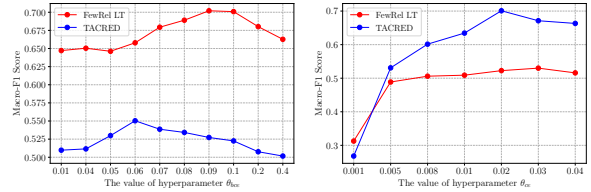


Figure 4: Performance with different hyperparameter settings.

## 6.3 Hyperparameter Analysis

In this section, we study the effects of reliability threshold  $r_h$  and  $r_m$  on optimization. Their values are given by the values of the elements ranked  $\theta_{ce}\%$  and  $\theta_{bce}\%$  from small to large. From Figure 4 it is possible to see that: (1) When  $\theta_{ce}$  and  $\theta_{bce}$  gradually increase from a small value, more training data are used for model optimization, and the performance of the model gradually improves. (2) When the value exceeds a certain threshold, further increasing the  $\theta_{ce}$  and  $\theta_{bce}$  will introduce more errors into the optimization, which degrades the performance of the model. (3) Compared with  $\mathcal{L}_{bce}$ ,  $\mathcal{L}_{ce}$  loss will make the model converge faster, so its optimal threshold of  $\mathcal{L}_{ce}$  should be less than that of  $\mathcal{L}_{bce}$  to prevent the overfitting of wrong labels.

## 7 Conclusions

In this work, we present a new setting, named actively supervised clustering for OpenRE, which provides the necessary guidance for clustering without a significant increase in human efforts. Along with this setting, a labeling strategy tailored for clustering is proposed, maximizing the clustering performance while discovering as many relations as possible. Different loss functions are assigned to pseudo labels with different reliability, which mitigate the risk of over-fitting to noise in pseudo labels. Experimental results show that this method significantly outperforms the existing two-stage methods for OpenRE.



## Limitations

Considering that the golden labels of all instances have been given in the datasets, we directly use these labels as manual labels without performing the manual labeling process. The practice implicitly assumes that all the manual labels are correct. However, with the increase of labeling scale, problems such as (1) inconsistent labeling granularity across annotators and (2) noise in manual labels gradually emerge. How to effectively improve the labeling quality and the robustness of the clustering model are worthy of attention.

## Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No.62076069,62206057,61976056), Shanghai Rising-Star Program (23QA1400200), and Natural Science Foundation of Shanghai (23ZR1403500).

## References

- Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019a. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*.
- Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019b. [Deep batch active learning by diverse, uncertain gradient lower bounds](#). *CoRR*, abs/1906.03671.
- Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does BERT look at? an analysis of BERT’s attention](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.
- Hady Elsahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frederique Laforest. 2017. Unsupervised open relation extraction. In *European Semantic Web Conference*, pages 12–16. Springer.
- Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR.
- Yoav Goldberg. 2019. Assessing bert’s syntactic abilities. *ArXiv*, abs/1901.05287.
- 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.
- Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip Yu. 2020. [SelfORE: Self-supervised relational feature learning for open relation extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3673–3682, Online. Association for Computational Linguistics.
- Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. [What does BERT learn about the structure of language?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.
- Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 1148–1158.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Diego Marcheggiani and Ivan Titov. 2016. [Discrete-State Variational Autoencoders for Joint Discovery and Factorization of Relations](#). *Transactions of the Association for Computational Linguistics*, 4:231–244.
- Dan Roth and Kevin Small. 2006. Margin-based active learning for structured output spaces. In *Machine Learning: ECML 2006*, pages 413–424, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pages 593–607, Cham. Springer International Publishing.
- Ozan Sener and Silvio Savarese. 2018. [Active learning for convolutional neural networks: A core-set approach](#).
- Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Dan Wang and Yi Shang. 2014a. [A new active labeling method for deep learning](#). In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 112–119.

Dan Wang and Yi Shang. 2014b. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE.

Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2019. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 219–228.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. [Structured relation discovery using generative models](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. *arXiv preprint arXiv:1704.06194*.

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.

Jun Zhao, Tao Gui, Qi Zhang, and Yaqian Zhou. 2021. A relation-oriented clustering method for open relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 219–228.

Jun Zhao, Xin Zhao, WenYu Zhan, Tao Gui, Qi Zhang, Liang Qiao, Zhanzhan Cheng, and Shiliang Pu. 2022. [Read extensively, focus smartly: A cross-document semantic enhancement method for visual documents NER](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2034–2043, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

## A Appendix

### A.1 Compared Active Labeling Strategy

To show the superiority of the proposed labeling strategy, the actively supervised clustering is combined with the following classical active learning strategies for comparison. **RANDOM** The naive baseline of randomly selecting  $k$  samples to query labels.

**CONFIDENCE** (Wang and Shang, 2014b) An uncertainty-based active learning algorithm that select  $k$  samples with smallest predicted class probability  $\max\{f_{\theta}(x)_i\}_{i=1,\dots,C}$

**MARGIN** (Roth and Small, 2006) An uncertainty-based active learning algorithm that selects the bottom  $k$  sorted according to the example’s multiclass margin, defined as  $f_{\theta}(x)_{\hat{y}} - f_{\theta}(x)_{y'}$ , where  $\hat{y}$  and  $y'$  are the indices of the largest and second largest entries of  $f_{\theta}(x)$ .

**ENTROPY**(Wang and Shang, 2014a) An uncertainty-based active learning algorithm that selects the top  $k$  samples according to the entropy of the sample’s class distribution.

**GRADIENT**(Ash et al., 2019b) A loss based active learning algorithm. The uncertainty is measured as the gradient magnitude with respect to parameters in the output layer.

### A.2 Additional Results

In this section, more detailed experimental settings and experimental results are given. Specifically, in TACRED dataset, 21 relations are held out as open relations to be discovered. In the underestimation (resp. overestimation) setting, we assume that the number of clusters is 10 (resp. 40). In FewRel and FewRel-LT datasets, 40 relations are held out as open relations and we assume that the number of clusters is 20 (resp. 80) for underestimation (resp. overestimation) setting. The results of the two settings are listed in tab. 4. When  $K$  is underestimated, it is observed that the precision of  $B^3$  is far lower than recall, which indicates that the model tends to mix multiple relations in the same cluster. When  $K$  is overestimated, the recall is far lower than precision, which indicates that the same relation tends to be clustered into multiple subclusters. Although the  $F_1$  of  $B^3$  metric seems to be tolerable, such imbalance clustering assignments cause great difficulties in relation labeling. If a cluster contains more than one relation, labeling the cluster as any relation will lead to the misidentification of other relations. If

a relation is clustered into multiple sub-clusters, the annotators have to label the same relation repeatedly, which leads to a significant increase in labeling costs.






























Dataset	Method	Setting	$B^3$			V-measure			ARI	Classification		
			Prec.	Rec.	$F_1$	Hom.	Comp.	$F_1$		Prec.	Rec.	$F_1$
TACRED	RW-HAC  (Elsahar et al., 2017)	U	0.317	0.668	0.430	0.443	0.668	0.532	0.291	0.244	0.246	0.171
	SelfORE  - (Hu et al., 2020)	U	0.339	0.759	0.469	0.468	0.809	0.593	0.412	0.121	0.244	0.160
	SelfORE  + (Hu et al., 2020)	U	0.575	0.251	0.349	0.680	0.522	0.591	0.290	0.469	0.481	0.231
	SelfORE  (Hu et al., 2020)	U	0.517	0.441	0.476	0.631	0.600	0.615	0.434	0.343	0.396	0.360
	RSN  (Wu et al., 2019)	P	0.312	0.807	0.451	0.445	0.768	0.563	0.354	0.149	0.118	0.225
	RoCORE  - (Zhao et al., 2021)	P	0.361	0.930	0.520	0.525	0.903	0.664	0.447	0.116	0.247	0.152
	RoCORE  + (Zhao et al., 2021)	P	0.714	0.530	0.608	0.805	0.698	0.748	0.552	0.612	0.649	0.311
	RoCORE  (Zhao et al., 2021)	P	0.696	0.685	0.690	0.786	0.786	0.787	0.640	0.547	0.594	0.563
	<b>Ours</b> 	A	0.742	0.821	<b>0.780</b>	0.807	0.856	<b>0.831</b>	<b>0.781</b>	0.698	0.715	<b>0.699</b>
	FewRel	RW-HAC  (Elsahar et al., 2017)	U	0.175	0.367	0.237	0.357	0.463	0.403	0.108	0.251	0.264
SelfORE  - (Hu et al., 2020)		U	0.365	0.710	0.482	0.620	0.800	0.699	0.368	0.282	0.442	0.327
SelfORE  + (Hu et al., 2020)		U	0.639	0.400	0.492	0.793	0.681	0.733	0.492	0.733	0.744	0.365
SelfORE  (Hu et al., 2020)		U	0.527	0.552	0.539	0.728	0.736	0.732	0.517	0.604	0.632	0.600
RSN  (Wu et al., 2019)		P	0.174	0.640	0.274	0.389	0.659	0.489	0.173	0.112	0.239	0.134
RoCORE  - (Zhao et al., 2021)		P	0.446	0.901	0.600	0.701	0.922	0.797	0.448	0.320	0.476	0.358
RoCORE  + (Zhao et al., 2021)		P	0.824	0.656	0.730	0.896	0.814	0.853	0.739	0.882	0.881	0.439
RoCORE  (Zhao et al., 2021)		P	0.806	0.843	<b>0.824</b>	0.883	0.896	0.889	<b>0.807</b>	0.827	0.868	0.837
<b>Ours</b> 		A	0.799	0.841	0.820	0.888	0.901	<b>0.894</b>	0.801	0.832	0.862	<b>0.838</b>
FewRel-LT		RW-HAC  (Elsahar et al., 2017)	U	0.255	0.322	0.285	0.379	0.421	0.399	0.145	0.190	0.176
	SelfORE  - (Hu et al., 2020)	U	0.266	0.633	0.374	0.466	0.676	0.552	0.290	0.079	0.154	0.099
	SelfORE  + (Hu et al., 2020)	U	0.641	0.298	0.407	0.771	0.589	0.668	0.370	0.589	0.647	0.305
	SelfORE  (Hu et al., 2020)	U	0.563	0.456	0.504	0.717	0.661	0.687	0.377	0.439	0.526	0.462
	RSN  (Wu et al., 2019)	P	0.211	0.500	0.297	0.350	0.510	0.415	0.193	0.098	0.173	0.117
	RoCORE  - (Zhao et al., 2021)	P	0.382	0.858	0.528	0.571	0.873	0.691	0.400	0.123	0.217	0.151
	RoCORE  + (Zhao et al., 2021)	P	0.714	0.530	0.608	0.805	0.698	0.748	0.552	0.612	0.649	0.311
	RoCORE  (Zhao et al., 2021)	P	0.662	0.717	0.689	0.800	0.801	0.800	0.581	0.507	0.538	0.517
	<b>Ours</b> 	A	0.650	0.845	<b>0.735</b>	0.790	0.885	<b>0.835</b>	<b>0.676</b>	0.530	0.609	<b>0.550</b>

Table 4: Main results on three relation extraction datasets.  and  represent that the number of relation types in unlabeled data is correctly and incorrectly estimated, respectively. In addition, - and + denotes the underestimation and overestimation, respectively.