HCL-TAT: A Hybrid Contrastive Learning Method for Few-shot Event Detection with Task-Adaptive Threshold

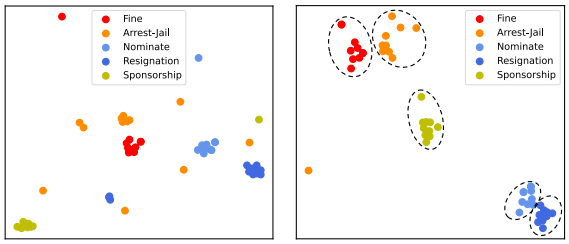
Ruihan Zhang[1,2], **Wei Wei**[1,2]*, **Xian-Ling Mao**[3], **Rui Fang**[4], **Dangyang Chen**[4]
[1]Cognitive Computing and Intelligent Information Processing (CCIIP) Laboratory,
School of Computer Science and Technology, Huazhong University of Science and Technology
[2]Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL)
[3]Department of Computer Science and Technology, Beijing Institute of Technology
[4]Ping An Property & Casualty Insurance company of China, Ltd
ruihanzhang@hust.edu.cn, weiw@hust.edu.cn, maoxl@bit.edu.cn
fangrui051@pingan.com.cn, chendangyang273@pingan.com.cn

Abstract

Conventional event detection models under supervised learning settings suffer from the inability of transfer to newly-emerged event types owing to lack of sufficient annotations. A commonly-adapted solution is to follow a identify-then-classify manner, which first identifies the triggers and then converts the classification task via a few-shot learning paradigm. However, these methods still fall far short of expectations due to: (i) insufficient learning of discriminative representations in low-resource scenarios, and (ii) trigger misidentification caused by the overlap of the learned representations of triggers and non-triggers. To address the problems, in this paper, we propose a novel **H**ybrid **C**ontrastive **L**earning method with a **T**ask-**A**daptive **T**hreshold (abbreviated as HCL-TAT), which enables discriminative representation learning with a two-view contrastive loss (*support-support* and *prototype-query*), and devises a easily-adapted threshold to alleviate misidentification of triggers. Extensive experiments on the benchmark dataset FewEvent demonstrate the superiority of our method to achieve better results compared to the state-of-the-arts. All the code and data of this paper will be available for online public access.

1 Introduction

Event detection (ED) is the subtask of information extraction (IE) (Pan et al., 2021, 2022), which aims at extracting events of task-specified types from an input text and is crucial for many downstream applications such as text summarization (Ge et al., 2016) and machine reading comprehension (Qiu et al., 2019). For example, in the sentence "Kenyan police intensify manhunt for ***terror*** suspects", an ideal ED model is to identify "***terror***" as a trigger word and classify it into "**Conflict.Attack**" event type. Previous works commonly formulate ED as a token-level classification problem and follow a supervised manner (Chen et al., 2015; Liu et al., 2017; Nguyen and Grishman, 2018; Zhao et al., 2018; Wang et al., 2019; Tong et al., 2020). Despite the promising results, in real-world scenarios, new event types are constantly emerging with only a few samples, which gives rise to the problem of low generation on newly-emerged event types, in view of insufficient annotated data. Therefore, it is reasonable to convert the traditional supervised-based event detection task to the few-shot event detection (FSED) problem.



(a) PA-CRF (b) HCL-TAT

Figure 1: Visualization of triggers in the same episode on FewEvent test set. The left and right half shows support set representations without and with hybrid contrastive learning, respectively.

Most works in FSED exploit metric-based meta-learning methods, in which the model needs to learn meta-knowledge from only a few instances in the *support set* and generalize to predict labels for instances in the *query set*. Specifically, in each episode, these methods first build a prototype for each class over the support set, then predict the labels for each instance in query set by matching with the closest prototype representation in the metric space. Indeed, many of these methods follow

*Corresponding author: Wei Wei.

an identify-to-classify paradigm and are capable of keeping high generation on new event type based on a few observed samples (Deng et al., 2020; Lai et al., 2020a,b). However, the performance of such two-stage model is limited to the problem of error propagation caused by trigger misidentification. Thus, recent works formulate FSED as a few-shot sequence labeling task and apply Conditional Random Field based (CRF-based) methods to jointly identify and classify triggers. Nevertheless, it is usually challenging for these CRF-based methods to fully learn event type dependencies from a single sentence, since a sentence commonly contains only one trigger, or even worse with limited data in FSED. An example is presented in Figure 1a, even if the state-of-the-art method PA-CRF (Cong et al., 2021) is powerful in exploring event type dependencies via Gaussian distribution for approximation, it is still incapable of generating well discriminative representations for each event type in the whole embedding space.

Actually, it is non-trivial to solve the problem. Typical metric-based models are mostly optimized with a query-anchored cross-entropy loss based on the similarity between each query instance and all prototypes. However, we find that the gradient directions of prototypes are not always optimized very well during training, and may not guarantee the learnt representations of prototypes are discriminative via optimizing the query-anchored loss, which in turn harms the learning of query instances. Detailed proof can be found in Appendix C.

To address the problem, in this paper, we propose a **H**ybrid **C**ontrastive **L**earning (**HCL**) method to improve the representations for instances in both support set and query set. Specifically, we first propose a **S**upport-**S**upport **C**ontrastive **L**earning (**SSCL**) method to makes it prone to generating more discriminative representations of prototypes, via encouraging instances of the same type closer and others of different types farther away within support set. However, the optimization direction of SSCL is still unclear and only information within the support set is considered. Therefore, inspired by (Gao et al., 2021b), we further design a **P**rototype-**Q**uery **C**ontrastive **L**earning (**PQCL**) method, which pulls query instances together with the prototype of the same type and pushs them apart from prototypes of different types. Logically, PQCL is capable of guiding the learning process of SSCL in a theoretically sound way, and such two contrastive learning losses are helpful for query-anchored loss to generate more discriminative representations for both the support set and the query set.

Additionally, another serious problem for typical FSED is that most of the words in ED sentences are non-triggers, *i.e.,* they do not belong to any event type, called "O" type. The unbalanced nature influences the representation learning of triggers, and leads to representation overlap problem between triggers and non-triggers. To address this issue, we further design a **T**ask-**A**daptive **T**hreshold (**TAT**). Our key insight is that, if the similarity between a query instance and all event types are lower than the average similarity between query instances and "O" type, then it is most likely to be a non-trigger. Thus, in each episode, we take the average similarity between all query instances and the prototype of "O" type as the threshold to filter misidentified triggers. This threshold can be easily adapted to any episode and ensure the generalization ability. As illustrated in Figure 1b, by combining the above components, our **HCL-TAT** could learn more discriminative representations and make better predictions, thus boosting the performance.

The contributions of our work are summarized as follows:

- We conduct the study of the limitation of query-anchored cross-entropy loss, namely, indistinguishability in embedding space, which harms the metric-based classification between query instances and prototypes.
- We propose a hybrid contrastive learning framework for the FSED task, which consists of two components, namely, SSCL and PQCL, to jointly improve the learning of instances on both the support set and the query set, which in return enhance the performance of FSED. Besides, we also propose a task-adaptive task-adaptive threshold (TAT) to eliminate the misidentification of trigger words.
- Experiments on FewEvent dataset in different settings demonstrate the advantages of our proposed HCL-TAT over various strong baseline methods.

2 Related Work

Data-driven Event Detection. Early ED approaches rely on traditional machine learning models with handcrafted features to extract events (Ji

and Grishman, 2008; Liao and Grishman, 2010). With advances in deep learning, many research efforts have been dedicated to enhancing ED with different neural network architectures like **CNN-based** model (Chen et al., 2015; Nguyen and Grishman, 2016) and **RNN-based** model (Nguyen et al., 2016; Feng et al., 2016). Recently, pre-trained language models (PLMs) are adopted to leverage rich information from large-scale corpora (Yang et al., 2019; Tong et al., 2020; Lai et al., 2020c). These methods achieve promising results for supervised ED, but they could not adapt well to FSED with limited data.

Few-shot Event Detection. In recent years, there have been several attempts to conduct meta-learning based approaches (Snell et al., 2017; Hu et al., 2018) for FSED. Deng et al. (2020) proposed the benchmark FSED dataset FewEvent, and designs a dynamic-memory-based prototypical network (DMBPN) to preserve contextual information of event mentions. Lai et al. (2020a) and Lai et al. (2020b) proposed intra and inter-cluster matching losses to provide more training signals. However, these works categorize events at sentence-level and are solely for the event classification task. Recently, Cong et al. (2021) proposed to treat ED as few-shot sequence labeling task and solved with a modified PA-CRF model, but they still suffer from poor representation learning. In this paper, we propose to exploit contrastive learning to produce more discriminative representations.

Contrastive Learning. Contrastive learning has been widely used in various domains based on an idea to pull together similar instances and push apart distinct instances. **Self-supervised learning** designs specific strategies to generate positive and negative pairs from unsupervised data (Chen et al., 2020; Gao et al., 2021a; Yan et al., 2021; Wang et al., 2021b; Zou et al., 2022a,b; Wang et al., 2022). As an extension, **supervised contrastive learning** which leverages label information to generate positive and negative pairs is further proposed (Khosla et al., 2020; Gunel et al., 2020; Wang et al., 2021a). Recently, contrastive learning is also used in few-shot learning tasks to improve the performance (Gao et al., 2021b), by pulling query samples closer to the prototype of the same class and further away from those of different classes. In this paper, we propose a hybrid contrastive learning (HCL) method, composed of two complementary contrastive losses, to generate more discriminative representations for both support and query set. With the help of a task-adaptive threshold (TAT) method, the proposed HCL-TAT can fully make use of limited data and boosts the performance.

3 Methodology

3.1 Overview

An overview of the proposed HCL-TAT method is illustrated in Figure 2. We first give the problem statement in Section 3.2, then introduce the backbone prototypical network in Section 3.3. In Section 3.4 and Section 3.5, we detailedly describe the hybrid contrastive learning and task-adaptive threshold method respectively. Finally, the training process is given in Section 3.6.

3.2 Problem Statement

In this paper, we formulate few-shot event detection (FSED) as a few-shot sequence labeling task. Given a sentence $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ composed of n tokens and its corresponding label sequence $\mathcal{Y} = \{y_1, y_2, ..., y_n\}$, each token x_i is categorized into an event type. Besides the defined event types in the dataset, we add an "O" type to denote for tokens that do not belong to any event type (called non-triggers). Then, FSED is defined with a typical N-way-K-shot setting. Specifically, given a support set $\mathcal{S} = \{\mathcal{X}^{(i)}, \mathcal{Y}^{(i)}\}_{i=1}^{N\times K}$ which has N event types with K labeled samples for each type, and a query set $\mathcal{Q} = \{\mathcal{X}^{(i)}, \mathcal{Y}^{(i)}\}_{i=1}^{N\times M}$ which has the same N event types as $\mathcal{S}$ with M samples for each type, we formulate a N-way-K-shot task or episode as $\mathcal{T} = \{\mathcal{S}, \mathcal{Q}\}$. Note that in each episode we also have an additional "O" type. The goal of FSED is to predict the labels for instances in $\mathcal{Q}$ based on $\mathcal{S}$. Specifically, in training stage, the classification results of $\mathcal{Q}$ are used to update model parameters, while in testing stage, the classification results are used to evaluate the model. The training phase is composed of a set of episodes $\mathcal{T}_{train} = \{\mathcal{T}_i\}_{i=1}^{M_{train}}$, and the testing phase is composed of another set of episodes $\mathcal{T}_{test} = \{\mathcal{T}_i\}_{i=1}^{M_{test}}$. M_{train} and M_{test} denote the number of training episodes and testing episodes respectively, and the label set of $\mathcal{T}_{train}$ and $\mathcal{T}_{test}$ are disjoint.

3.3 Prototypical Network

Following Cong et al. (2021), we select BERT as the text encoder to represent event mentions. We encode the sentence $\mathcal{X}$ into a sequence of hidden

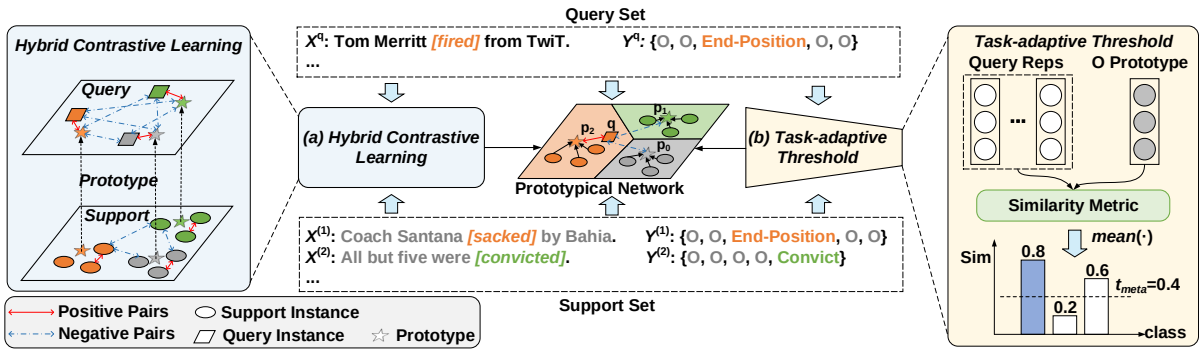


Figure 2: Overall framework of the proposed HCL-TAT model. HCL-TAT is based on a prototypical network, composed of two components: (a) hybrid contrastive learning including support-support contrastive learning and prototype-query contrastive learning; (b) task-adaptive threshold based on the logits in each episode.

embeddings as follows,

$$\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\} = f(\mathcal{X}, \theta), \tag{1}$$

where $f(:, \theta)$ is the encoder and $\mathbf{h}_i$ is the hidden representation of each token x_i.

We then adopt the widely-used prototypical network (Proto) (Snell et al., 2017) as our backbone, in which the classification results are obtained by matching instances with prototypes of each class in the metric space. During each episode, Proto first computes prototypes for each class c by averaging instances in support set:

$$\mathbf{p}_c = \frac{1}{K} \sum_{i \in \mathcal{S}(c)} \mathbf{h}_i, \ c = 0, 1, ..., N, \tag{2}$$

where $\mathbf{p}_i$ is the prototype for class c, $\mathcal{S}_c$ represents for all the words of class c in $\mathcal{S}$. Then in training phase, the prototypes and query instances are fed into a non-parametric distance-based classifier to compute the cross-entropy loss for few-shot classification:

$$\mathcal{L}_{CE} = - \sum_{(x_i, y_i) \in \mathcal{Q}} \log P(y_i | x_i, \mathcal{S}), \tag{3}$$

$$P(y_i | x_i, \mathcal{S}) = \frac{\exp(-d(\mathbf{h}_i, \mathbf{p}_{y_i}))}{\sum_{c \in \mathcal{C}} \exp(-d(\mathbf{h}_i, \mathbf{p}_c))}, \tag{4}$$

where $\mathcal{C} = \{0, 1, ..., N\}$ denotes for the sampled class set including an additional "O" type, $d(\cdot)$ is a metric function to measure the similarity between query instances and prototypes.

We then analyze the bottleneck of query-anchored loss in Eq. (3). Assuming $d(\cdot)$ is dot product, $\mathbf{p}^{pos}$ represents for the prototype of class y_i and $\mathbf{p}^n$ represents for any prototype of different classes, we compute the gradient *w.r.t.* $\mathbf{h}_i$, $\mathbf{p}^n$ and $\mathbf{p}^{pos}$ respectively:

$$\frac{\partial \mathcal{L}_{CE}}{\partial \mathbf{h}_i} = \frac{\sum_n \Delta_n (\mathbf{p}^n - \mathbf{p}^{pos})}{1 + \sum_n \Delta_n}, \tag{5}$$

$$\frac{\partial \mathcal{L}_{CE}}{\partial \mathbf{p}^n} = \frac{\Delta_n \mathbf{h}_i}{1 + \sum_n \Delta_n}, \frac{\partial \mathcal{L}_{CE}}{\partial \mathbf{p}^{pos}} = - \frac{\sum_n \Delta_n \mathbf{h}_i}{1 + \sum_n \Delta_n}, \tag{6}$$

$$\Delta_n = \exp(\mathbf{h}_\mathrm{i} \cdot \mathbf{p}^\mathrm{n} \text{-} \mathbf{h}_\mathrm{i} \cdot \mathbf{p}^\mathrm{pos}) \, . \tag{7}$$

We can conclude that the query instance $\mathbf{h}_i$ has a better update direction compared with prototypes, and this makes it hard to learn good representations for the prototypes, which could in turn affects the learning of $\mathbf{h}_i$, since the gradient *w.r.t.* $\mathbf{h}_i$ approaches to zero when $\mathbf{p}^{pos}$ and $\mathbf{p}^n$ are close in the embedding space. Therefore, we propose hybrid contrastive learning method to learn more discriminative representations and break this bottleneck. Proof details can be found in Appendix C.

3.4 Hybrid Contrastive Learning

Hybrid contrastive learning (HCL) consists of two components: support-support contrastive learning (SSCL) applied to instances within $\mathcal{S}$ for more discriminative prototype representations and prototype-query contrastive learning (PQCL) applied between $\mathcal{S}$ and $\mathcal{Q}$ to guide the optimization process for instances in both set.

3.4.1 Support-Support Contrastive Learning

To produce better prototypes, we should learn more discriminative representations for instances in $\mathcal{S}$. Therefore, naturally we follow a supervised contrastive learning manner to construct our SSCL loss. We use label information to construct positive and negative pairs. In each episode, for word x_i, we

take instances of its same class in $\mathcal{S}$ as positive pairs $\mathcal{P}(i)$ and instances of different classes in $\mathcal{S}$ as negative pairs $\mathcal{N}(i)$. Empirically, since a non-linear projection layer improves the representation quality for contrastive learning (Chen et al., 2020), we exploit a 2-layer MLP to project the token representations to a latent space.

$$\tilde{\mathbf{h}}_i = \mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{h}_i), \tag{8}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are trainable weights, and σ is an activation function.

We then encourage the positive pairs to pull closer and negative pairs to push further. In this way, we make instances in $\mathcal{S}$ more discriminative and produce more compact clusters to generate better prototypes. Specifically, we take dot product to measure the similarity between instances, and the SSCL loss is calculated as follows:

$$\mathcal{L}_{SSCL} = \sum_{(x_i,y_i)\in\mathcal{S}} \mathcal{L}_{SSCL_i}, \tag{9}$$

$$\mathcal{L}_{SSCL_i} = -\log \frac{\exp(\tilde{\mathbf{h}}_i \cdot \tilde{\mathbf{h}}_j/\tau)}{\sum_{k\neq i}\exp(\tilde{\mathbf{h}}_i \cdot \tilde{\mathbf{h}}_k/\tau)}, \tag{10}$$

where τ is a scalar temperature parameter. In practice, we apply ℓ_2 normalization to $\tilde{\mathbf{h}}_i$ for numerical stability.

3.4.2 Prototype-Query Contrastive Learning

By adopting SSCL, instances in the support set are enforced to be more separable. However, the optimization direction is still unclear without the guidance of query set. To this end, inspired by (Gao et al., 2021b), we propose prototype-query contrastive learning (PQCL) to eliminate the gap. Similarly, we take each prototype as the anchor, and regard query instances of the same class as positive pairs and query instances of different classes as negative pairs. Given a class $c \in \mathcal{C}$, the positive pairs set and negative pairs set are defined as $\mathcal{Q}_c^{pos}$ and $\mathcal{Q}_c^{neg}$ respectively. The PQCL loss is calculated by:

$$\mathcal{L}_{PQCL} = \sum_{c\in\mathcal{C}} \sum_{(x_i,y_i)\in\mathcal{Q}_c^{pos}} \mathcal{L}_{PQCL_c^i}, \tag{11}$$

$$\mathcal{L}_{PQCL_c^i} = -\log \frac{sim_c^i}{sim_c^i + \sum_{(x_k,y_k)\in\mathcal{Q}_c^{neg}} sim_c^k}, \tag{12}$$

$$sim_c^i = \exp(\mathbf{p}_c \cdot \tilde{\mathbf{h}}_i/\tau). \tag{13}$$

We then combine the two contrastive losses as a hybrid contrastive learning (HCL) method, to produce more discriminative prototypes for query instances. Our HCL is complementary to the query-anchored cross-entropy loss $\mathcal{L}_{CE}$, and by optimizing the losses jointly, the model can obtain better representations for both support set and query set.

3.5 Task-adaptive Threshold

The proposed HCL could produce better representations for both support set and query set. However, since "O" type contains all instances that do not belong to any event type, the semantics of those instances are complex, making it hard to separate them even with the help of HCL. Therefore, it's important to further regularize the classification process for more accurate predictions. Naturally, we can set a threshold and let query instances whose similarity to all event types are lower than the threshold to output an "O" type prediction.

However, manually designed thresholds are hard to tune and lack generalization. Thus, based on the episodic training in FSED, we propose a task-adaptive threshold method. Intuitively, since the similarity between query instance and prototype can be regarded as a confidence score, in each episode, we take the average value of similarities between all query instances and "O" type as the threshold for automatically separating triggers and non-triggers. If the similarity between a query instance and an event type cannot even reach the calculated threshold, then it's unlikely for the query instance to be a trigger of that event type. We use probability to represent for similarity and the threshold t can be defined as:

$$t_{meta} = \frac{1}{|\mathcal{Q}|} \sum_{(x_i,y_i)\in\mathcal{Q}} P(y_i = 0|x_i, \mathcal{S}). \tag{14}$$

In this way, we could regularize the classification process by eliminating the misidentification of triggers.

3.6 Training Process

By combining the hybrid contrastive loss and task-adaptive threshold, we obtain the full method HCL-TAT. In each training episode, the model is optimized with the mixture of query-anchored cross-entropy loss and hybrid contrastive loss:

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha\mathcal{L}_{SSCL} + \beta\mathcal{L}_{PQCL}, \tag{15}$$

where α and β are trade-off parameters to balance the losses. During training and testing phase, we

both use the threshold to regularize the classification. Finally, we randomly select multiple testing episodes and report the mean result to evaluate the performance.

4 Experiments

4.1 Experimental Settings

Dataset. We conduct experiments on the newly-proposed largest few-shot event dataset **FewEvent**[1] proposed by (Deng et al., 2020). FewEvent contains more than 70000 event instances over 100 fine-grained event types, and is built from ACE2005, KBP2017 and external knowledge bases like Freebase and Wikipedia, thus it's representative and can take coverage of most event types. Following (Cong et al., 2021), we use the same 80 event types for training, 10 event types for validation and the rest 10 event types for testing, where event types between each subset are disjoint. The statistics of FewEvent dataset is listed in Appendix A.

Evaluation. Same as previous works (Chen et al., 2015), we report micro precision, recall and F1 score over event types, among which F1 score is the most important metric. Following (Cong et al., 2021), we use the same episodic evaluation method to evaluate our model in few-shot settings, by randomly selecting episodes containing N-way-K-shot samples from the test set. We run each experiment 5 times to get the averages and standard deviations for fair comparison.

Implementation Details. We use bert-base-uncased model through huggingface's transformers[2] to obtain the 768-dimensional token-level embeddings. Following (Cong et al., 2021), we also use episodic training and evaluation methods for few-shot event detection. In the training stage, we randomly select N-way-K-shot samples from training set for 20000 iterations. In each iteration, we first randomly select N event types then assign K instances for each type as the support set, and select other M samples for each type as the query set. We manually tune the hyper-parameters by running similar N-way-K-shot episodic evaluation on the validation set for 1000 iterations, and the final test result is obtained by 3000 iterations on the test set. Since only 0.12% of sentences are longer than 128, we set the maximum length to 128 and pad sentences shorter than 128 with a special character "[PAD]". We optimize our model using AdamW with a 1e-5 learning rate. For the contrastive learning part, the scalar temperature parameter τ is set to 0.5 and 0.1 for SSCL and PQCL, respectively, and the trade-off parameters of α and β are both set to 0.5. We experiment with Pytorch 1.10 in Ubuntu 18.04. For 5-way-5-shot, 5-way-10-shot and 10-way-5-shot settings, we run the experiments on one NVIDIA GeForce RTX 2080 Ti. For 10-way-10-shot setting, we run the experiments on one NVIDIA GeForce RTX 3090.

[1]https://github.com/231sm/Low_Resource_KBP

[2]https://github.com/huggingface/transformers

4.2 Baselines

To comprehensively investigate the effectiveness of our method, we choose a variety of baselines on FewEvent, including two-stage methods and unified methods.

For **two-stage models**, we compare with **LoLoss** (Lai et al., 2020a), **MatchLoss** (Lai et al., 2020b) and **DMBPN** (Deng et al., 2020), by adding a trigger identification model before their methods. LoLoss uses matching information between examples in the support set as additional training signals, MatchLoss extends LoLoss to consider intra-cluster matching and inter-cluster information. DMBPN proposes the benchmark FSED dataset FewEvent, and proposes a dynamic memory network based model to preserve more contextual information for FSED.

For **unified models**, we compare with three typical few-shot classification baselines, **Proto-dot**, **Match** and **Proto**. Besides, we also compare with the state-of-the-art CRF-based methods, **Vanilla CRF**, **CDT** (Hou et al., 2020) and **PA-CRF** (Cong et al., 2021). Match, Proto and Proto-dot are similar metric-based model, with cosine similarity, euclidean distance and dot product used as metric respectively. We re-implement the three baselines since Cong "[PAD]" embeddings are used to calculate prototype of "O" type in previous work (Cong et al., 2021). Vanilla CRF adds a simple CRF layer behind baseline models, while CDT and PA-CRF exploits Gaussian distribution approximation and collapsed dependency transfer to enhance the CRF layer, respectively.

4.3 Overall Performance

We use Proto as the backbone model for HCL-TAT. Table 1 show the overall performance on FewEvent test set. We can observe that: (i) Compared with two-stage models, unified models achieve much

Model	5-way-5-shot	5-way-10-shot	10-way-5-shot	10-way-10-shot
LoLoss	31.51 ± 1.56	31.70 ± 1.21	30.46 ± 1.38	30.32 ± 0.89
MatchLoss	30.44 ± 0.99	30.68 ± 0.78	28.97 ± 0.61	30.05 ± 0.93
DMBPN	37.51 ± 2.60	38.14 ± 2.32	34.21 ± 1.45	35.31 ± 1.69
Proto-dot†	41.54 ± 3.82	42.21 ± 0.68	33.27 ± 2.37	39.23 ± 2.95
Match†	30.09 ± 1.71	48.10 ± 1.38	28.94 ± 1.15	45.91 ± 1.98
Proto†	47.30 ± 2.55	54.81 ± 2.27	42.48 ± 1.00	50.14 ± 0.65
Vanilla CRF	59.01 ± 0.81	62.21 ± 1.94	56.00 ± 1.51	59.35 ± 1.09
CDT	59.30 ± 0.23	62.77 ± 0.12	56.41 ± 1.09	59.44 ± 1.83
PA-CRF	62.25 ± 1.42	64.45 ± 0.49	58.48 ± 0.68	61.64 ± 0.81
HCL-TAT	**66.96** ± 0.70	**68.80** ± 0.85	**64.19** ± 0.96	**66.00** ± 0.81

Table 1: F1 scores (10^{-2}) of evaluated methods on FewEvent test set. † means the model is re-implemented by ourselves. The best scores are highlighted in boldface, with $p < 0.02$ under t-test.

better results, which demonstrates the influence of error propagation and the advantage of unifed architectures. (ii) Among the three metric-based few-shot classification baselines, Proto achieves the best result. This indicates that euclidean distance is the best metric for FSED. Note that this observation conflicts with the conclusion in PA-CRF, because the authors uses "[PAD]" embeddings to calculate the prototype of "O" type. (iii) The re-implemented baselines have a lower performance compared with those in (Cong et al., 2021). This implies that when optimizing with only query-anchored loss, the model cannot learn discriminative representations and even using "[PAD]" embeddings to calculate the prototype of "O" type is a better choice than averaging the corresponding instance representations. (iv) Our proposed HCL-TAT achieves new state-of-the-art results under all four settings. Specifically, compared with PA-CRF, HCL-TAT brings an improvement of 4.71%, 4.35%, 5.71% and 4.36%, which proves the effectiveness of HCL-TAT to learn better representations and obtain better classification results. (v) Our proposed HCL-TAT achieve even higher results in 5-shot settings, which demonstrates the ability of HCL-TAT to make full use of limited data compared with other methods.

4.4 Ablation Study

We conduct ablation studies to investigate the effectiveness of each component in the proposed HCL-TAT model. The experimental results are shown in Table 2.

Effect of Contrastive Learning. We remove the two contrastive losses in HCL to observe the performance. We can conclude that: (i) When removing SSCL, the F1 score drops 1.93%/2.38% under 5-way-5-shot and 5-way-10-shot settings respectively, which indicates that SSCL could benefit the model by producing more separable representations for prototypes. (ii) When removing PQCL, the F1 scores drop by 3.11%/2.59%. This shows that compared with SSCL, PQCL contributes more to the improvement due to the information interaction between support and query set. Especially, in 5-way-10-shot-setting, the performance gap between SSCL and PQCL is slight, we believe the reason is that more instances in support set provide more training signals for SSCL. (iii) When removing both SSCL and PQCL, *i.e.,* HCL, the F1 scores drop significantly by 7.58%/4.23%, which proves the effectiveness of combining the two contrastive losses. Besides, we note that HCL achieves higher improvement in more difficult setting (5-shot), which further proves the superiority of HCL in few-shot scenarios. (iv) The improvement of HCL mainly comes from the improvement of precision score, which indicates that by producing more discriminative representations, HCL reduces the representation overlap between different types and thus alleviates the misidentification of triggers.

Effect of Task-adaptive Threshold. To prove the effect of task-adaptive threshold (TAT), we remove this module and evaluate the performance under 5-way-5-shot and 5-way-10-shot settings. The results show that without TAT, the F1 scores drop dramatically by 8.84%/8.71%, in which the preci-

Model	5-way-5-shot			5-way-10-shot		
	P	R	F1	P	R	F1
HCL-TAT	**62.63** ± 2.31	72.04 ± 1.93	**66.96** ± 0.70	**63.87** ± 2.35	74.65 ± 1.36	**68.80** ± 0.85
w/o SSCL	59.61 ± 2.48	71.65 ± 1.72	65.03 ± 0.82	60.22 ± 4.78	74.38 ± 1.81	66.42 ± 2.34
w/o PQCL	57.50 ± 1.80	71.88 ± 1.52	63.85 ± 0.67	60.88 ± 2.18	72.63 ± 1.23	66.21 ± 1.14
w/o HCL	49.52 ± 4.34	74.67 ± 3.36	59.38 ± 2.59	57.72 ± 2.72	73.35 ± 1.10	64.57 ± 1.69
w/o TAT	46.69 ± 1.25	**76.98** ± 0.29	58.12 ± 0.94	49.56 ± 1.11	76.33 ± 0.67	60.09 ± 0.92

Table 2: Precision, recall and F1 scores (10^{-2}) of ablation study results on FewEvent test set. When remove both HCL and TAT, the method degenerates to a Proto model.

sion scores contribute most, with a 15.94%/14.31% decrease. This demonstrates that TAT could on the other hand eliminate the misidentification of triggers by regularizing the classification process.

4.5 Performance of Trigger Identification

Model	FSTI	FSED
PA-CRF	63.68	62.25
HCL-TAT	68.18	66.96

Table 3: Average F1 scores (10^{-2}) of HCL-TAT and PA-CRF on FSTI and FSED tasks, on FewEvent test set under 5-way-5-shot setting.

To investigate the effectiveness of our method to solve trigger identification problem, we compare the results between HCL-TAT and the state-of-the-art method PA-CRF. As shown in Table 3, HCL-TAT achieves 68.18% on few-shot trigger identification (FSTI) task, bringing a +4.5% improvement compared with PA-CRF. This indicates that the main improvement of HCL-TAT comes from more accurate identification of trigger words, which can be attributed to two aspects. First, HCL-TAT learns more discriminative representations for both support set and query set through hybrid contrastive learning. Second, the introduced task-adaptive threshold further regularizes the classification process to avoid misidentification.

4.6 N-way-K-shot Evaluation

We conduct N-way-K-shot evaluation to investigate the performance tendency of different models in different few-shot settings. For fair comparison, we re-run PA-CRF with the released source code to obtain the results. As illustrated in Figure 3, generally, when the shot number K is fixed, F1 scores tend to drop with the increase of way number N, and when the way number N is fixed, F1

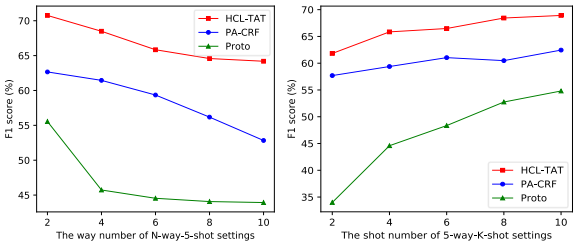
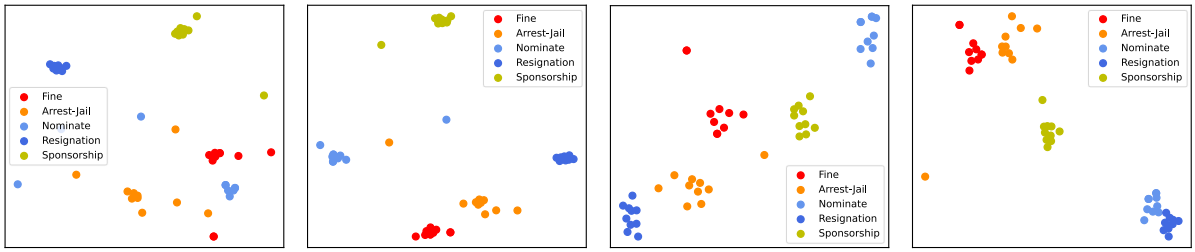


Figure 3: N-way-K-shot evaluations for three different models. The left part illustrates F1 scores in N-way-5-shot settings, and the right part illustrates F1 scores in 5-way-K-shot settings. We run each experiment once to analyze the tendency of F1 scores.

scores tend to improve with the increase of shot number K. Besides, we can observe that the backbone Proto model is more sensitive to the change of way and shot numbers, which indicates that the vanilla prototypical network is more vulnerable to limited data. Moreover, in all settings, HCL-TAT outperforms other methods with a large margin, especially when N is large. This proves that our method shows more robustness to the way number with the help of HCL to fully exploit information between event types and produce more discriminative representations.

4.7 Learning Visualization

Given the same test episode, we use t-SNE to visualize the embeddings of triggers for the backbone model Proto with and without using contrast learning. From Figure 4, we can observe that: (i) Proto achieves similar results as PA-CRF (see Figure 1a), with some instances hard to categorize. This proves that PA-CRF fails to improve representation learning by modeling label dependencies. (ii) SSCL and PQCL both contributes to the improvement of representations, while SSCL forms more compact clusters and PQCL learns more separable embedding space. (iii) When jointly using HCL, advan-



(a) Proto (b) Proto+SSCL (c) Proto+PQCL (d) Proto+HCL

Figure 4: Visualization of trigger embeddings in the same episode on FewEvent test set, under 5-way-10-shot setting. From left to right, the visualazation results of four FSED models are given respectively.

tages of SSCL and PQCL are combined and more discriminative representations are produced to improve the performance. (iv) Besides, in Figure 4d, "Fine" and "Arrest-Jail", "Nominate" and "Resignation" have the common supertype "Justice" and "Personnel", respectively, and representations of event types belonging to the same supertype are relatively close. This shows that HCL automatically learns the relations between event types.

5 Conclusion

In this paper, we propose a contrastive learning method to make full use of limited data and produce more discriminative representations for FSED. Specifically, we first propose a hybrid contrastive loss, composed of support-support contrastive loss and prototype-query loss, to conduct supervised contrastive learning within support set and between support and query set respectively. Furthermore, we design a task-adaptive threshold method, to regularize the distance-based classifier in each episode. Experimental results show that by improving the representation learning and classifier learning simultaneously, our method boost the performance under all four settings on FewEvent dataset.

Limitations

In FewEvent, each sentence has exactly one trigger word, so the sampling process is simplified into randomly selecting sentences containing specific events, and in each episode, it's unlikely for instances of "O" type to belong to any other event type that is not inluded in the current episode. However, in real-world scenarios, a sentence may contains multiple trigger words, which could bring more complicated settings. For example, we have to consider that instances of "O" type might belong to other event types that are not sampled in this episode, and the contrastive loss should be modified to adapt for such scenarios.

Besides, due to the huge memory cost, we only make full use of provided data, and do not consider data augmentation in contrastive learning, which has been proved effective in previous contrastive learning works. We believe that by conducting data augmentation and introducing more self-supervised signals, the performance of FSED could be further improved, which is worth for future research.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No.62276110, Grant No.61602197, Grant No.61772076, in part by CCF-AFSG Research Fund under Grant No.RF20210005, and in part by the fund of Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL). The authors would also like to thank the anonymous reviewers for their comments on improving the quality of this paper.

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL-IJCNLP*, pages 167–176.

Xin Cong, Shiyao Cui, Bowen Yu, Tingwen Liu, Yubin Wang, and Bin Wang. 2021. Few-shot event detection with prototypical amortized conditional random field. In *Proc. of Findings of ACL*.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *WSDM*.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *ACL*, pages 66–71.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021a. SimCSE: Simple contrastive learning of sentence embeddings. In *EMNLP*.

Yizhao Gao, Nanyi Fei, Guangzhen Liu, Zhiwu Lu, and Tao Xiang. 2021b. Contrastive prototype learning with augmented embeddings for few-shot learning. In *Uncertainty in Artificial Intelligence*, pages 140–150. PMLR.

Tao Ge, Lei Cui, Baobao Chang, Sujian Li, Ming Zhou, and Zhifang Sui. 2016. News stream summarization using burst information networks. In *EMNLP*, pages 784–794.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *ACL*, pages 1381–1393.

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. 2018. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *NeurIPS*, volume 33, pages 18661–18673.

Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2020a. Exploiting the matching information in the support set for few shot event classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 233–245. Springer.

Viet Dac Lai, Thien Huu Nguyen, and Franck Dernoncourt. 2020b. Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020c. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *EMNLP*, pages 5405–5411.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *ACL*, pages 789–797.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *ACL*, pages 1789–1798.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *NAACL*, pages 300–309.

Thien Huu Nguyen and Ralph Grishman. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *EMNLP*, pages 886–891.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*.

Weiran Pan, Wei Wei, and Xian-Ling Mao. 2021. Context-aware entity typing in knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2240–2250, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Weiran Pan, Wei Wei, and Feida Zhu. 2022. Automatic noisy label correction for fine-grained entity typing. In *Proceedings of the Thirty-first International Joint Conference on Artificial Intelligence, IJCAI-22*.

Delai Qiu, Yuanzhe Zhang, Xinwei Feng, Xiangwen Liao, Wenbin Jiang, Yajuan Lyu, Kang Liu, and Jun Zhao. 2019. Machine reading comprehension using structural knowledge graph-aware network. In *EMNLP-IJCNLP*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NeurIPS*, volume 30, pages 4077–4087.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *ACL*, pages 5887–5897.

Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang. 2021a. Contrastive learning based hybrid networks for long-tailed image classification. In *CVPR*, pages 943–952.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *NAACL*, pages 998–1008.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021b. CLEVE: Contrastive Pre-training for Event Extraction. In *ACL-IJCNLP*, pages 6283–6297.

Ziyang Wang, Huoyu Liu, Wei Wei, Yue Hu, Xian-Ling Mao, Shaojian He, Rui Fang, et al. 2022. Multi-level contrastive learning framework for sequential recommendation. *arXiv preprint arXiv:2208.13007*.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *ACL-IJCNLP*, pages 5065–5075.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *ACL*, pages 5284–5294.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *ACL*, pages 414–419.

Ding Zou, Xian-Ling Mao, Ziyang Wang, Minghui Qiu, Feida Zhu, and Xin Cao. 2022a. Multi-level cross-view contrastive learning for knowledge-aware recommender system. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2022, Madrid, Spain, July 11-15, 2022*.

Ding Zou, Wei Wei, Ziyang Wang, Xian-Ling Mao, Feida Zhu, Rui Fang, and Dangyang Chen. 2022b. Improving knowledge-aware recommendation with multi-level interactive contrastive learning. In *CIKM*.

A Dataset Statistics

Subset	#Class	#Trigger	#Avg.Len
Train	80	69088	36.5
Valid	10	2274	38.6
Test	10	748	30.8

Table 4: The statistics of FewEvent Dataset. #Class, #Trigger and #Avg.Len denotes the number of classes, the number of triggers and the average length of sentences in each split part respectively.

The statistics of FewEvent dataset are listed in Table 4, including the number of event types (#Class), the number of triggers (#Trigger) and the average length of sentences (#Avg.Len) of train, valid and test set. The length distribution of sentences in FewEvent is illustrated in Figure 5, showing that most of the sentence lengths are within 128. Thus we set the maximum length to 128.

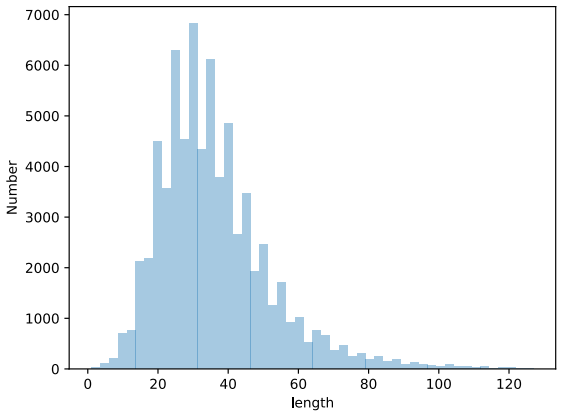


Figure 5: Length distribution of sentences in FewEvent dataset.

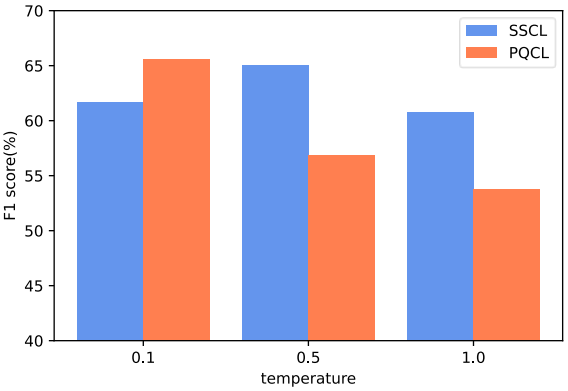


Figure 6: F1 scores(10^{-2}) over different temperature values on the two contrastive losses. The results are obtained under 5-way-5-shot setting in FewEvent test set.

B Parameter Study

We study the influence of the scalar temperature τ on our two contrastive losses, SSCL and PQCL. Figure 6 shows the results of the backbone model with only SSCL or PQCL, respectively. We can empirically observe that, PQCL benefits from smaller temperature ($\tau = 0.1$) and a larger temperature brings improvement for SSCL ($\tau = 0.5$).

C Bottleneck Analysis for Query-anchored Cross-entropy Loss

Theorem 1. *Optimized with Eq. (3) could not produce discriminative representations for prototypes, and further harms the representation learning of anchor query instance.*

Proof. For simplicity, we assume dot product is used as the similarity metric in Eq. (3). The loss

$\mathcal{L}_{CE}$ can thus be converted as follows:

$$\begin{aligned}\mathcal{L}_{CE} &= -\log \frac{\exp(\mathbf{h}_i \cdot \mathbf{p}^{pos})}{\exp(\mathbf{h}_i \cdot \mathbf{p}^{pos}) + \sum_n \exp(\mathbf{h}_i \cdot \mathbf{p}^{n})} \\ &= -\log \frac{1}{1 + \sum_n \frac{\exp(\mathbf{h}_i \cdot \mathbf{p}^{n})}{\exp(\mathbf{h}_i \cdot \mathbf{p}^{pos})}} \\ &= \log(1 + \sum_n \frac{\exp(\mathbf{h}_i \cdot \mathbf{p}^{n})}{\exp(\mathbf{h}_i \cdot \mathbf{p}^{pos})}) \\ &= \log(1 + \sum_n \exp(\mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{n}} \text{-} \mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{pos}})),\end{aligned} \tag{16}$$

where $\mathbf{h}_i$ is the representation of the anchor query instance, $\mathbf{p}^{pos}$ is the positive prototype of the same class, and $\mathbf{p}^{n}$ is a negative prototype of different classes. Then we calculate the partial derivative with respect to $\mathbf{h}_i$ to compute the gradient of $\mathbf{h}_i$. Let $\Delta_n = \exp(\mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{n}} \text{-} \mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{pos}})$, the calculation procedure is as follows:

$$\begin{aligned}\frac{\partial \mathcal{L}_{CE}}{\partial \mathbf{h}_i} &= \frac{\sum_n \exp(\mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{n}} \text{-} \mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{pos}}) \,|_{\mathbf{h}_i}}{1 + \sum_n \exp(\mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{n}} \text{-} \mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{pos}})} \\ &= \frac{\sum_n \exp(\mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{n}} \text{-} \mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{pos}})(\mathbf{p}^{n} - \mathbf{p}^{pos})}{1 + \sum_n \exp(\mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{n}} \text{-} \mathbf{h}_{\mathrm{i}} \cdot \mathbf{p}^{\mathrm{pos}})} \\ &= \frac{\sum_n \Delta_n (\mathbf{p}^{n} - \mathbf{p}^{pos})}{1 + \sum_n \Delta_n}.\end{aligned} \tag{17}$$

Similarly, we can compute the gradient of $\mathbf{p}^{n}$ and $\mathbf{p}^{pos}$ as follows:

$$\frac{\partial \mathcal{L}_{CE}}{\partial \mathbf{p}^{n}} = \frac{\Delta_n \mathbf{h}_i}{1 + \sum_n \Delta_n}, \tag{18}$$

$$\frac{\partial \mathcal{L}_{CE}}{\partial \mathbf{p}^{pos}} = -\frac{\sum_n \Delta_n \mathbf{h}_i}{1 + \sum_n \Delta_n}. \tag{19}$$

We can conclude that the update direction of $\mathbf{h}_i$ is better than $\mathbf{p}^{pos}$ and $\mathbf{p}^{n}$, so it's hard for prototypes to learn discriminative representations. Furthermore, if $\mathbf{p}^{pos}$ and $\mathbf{p}^{n}$ are close, then the gradient of $\mathbf{h}_i$ approaches to zero, making it hard to learn discriminative representation for the anchor query instance as well. □

Theorem 1 indicates that bad prototype representations bring bad query representations as well, and eventually harm the performance of FSED models. Therefore, we propose the contrastive learning based method to learn more discriminative representations for both support set and query set.