Mitigating Prototype Shift: Few-Shot Nested Named Entity Recognition with Prototype-Attention

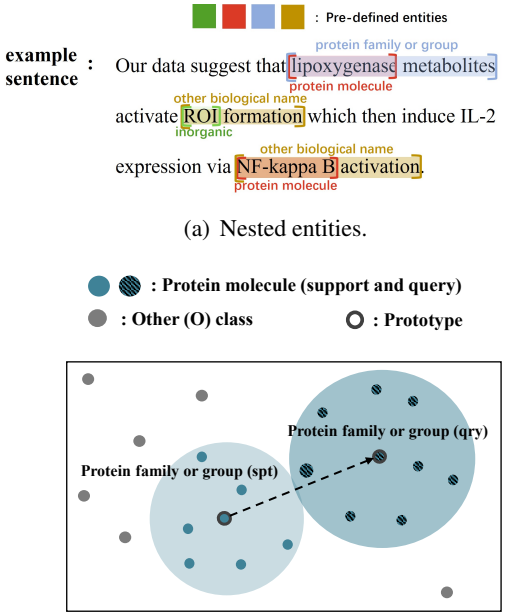
Anonymous ACL submission

Abstract

Nested entities are prone to obtain similar representations in pre-trained language models, posing challenges for Named Entity Recognition (NER), especially in the few-shot setting where prototype shifts often occur due to distribution differences between the support and query sets. In this paper, we regard entity representation as the combination of prototype and non-prototype representations. With a hypothesis that using the prototype representation specifically can help mitigate potential prototype shifts, we propose a Prototype-Attention mechanism in the Contrastive Learning framework (PACL) for the few-shot nested NER. PACL first generates prototype-enhanced span representations to mitigate the prototype shift by applying a prototype attention mechanism. It then adopts a novel prototype-span contrastive loss to reduce prototype differences further and overcome the O-type's non-unique prototype limitation by comparing prototype-enhanced span representations with prototypes and original semantic representations. Our experiments show that the PACL outperformed baseline models on the 1-shot and 5-shot tasks in terms of F_1 score. Furthermore, experiments on English datasets show the effectiveness of PACL, and experiments on cross-lingual datasets show the robustness of PACL. Further analyses indicate that our Prototype-Attention mechanism is a simple but effective method and exhibits good generalizability[1].

1 Introduction

The few-shot Named Entity Recognition (NER) task has gained a lot of attention in recent years as it aims to address the limitations of traditional NER methods that rely on a large number of labeled training instances, which can be both time-consuming and experience-dependent. This task deals with the NER problem using only a few labeled instances.



(a) Nested entities.

(b) Prototype shift

Figure 1: **(a)** Example of a sentence with nested entities from the GENIA dataset. **(b)** Illustration of prototype shifts, where the prototypes differ due to the distribution difference between the support and query sets.

Researchers have made significant progress on this task by applying deep learning models, including pre-trained-model-based (Florez and Mueller, 2019; Hou et al., 2019; Yang et al., 2021; Wang et al., 2022b), metric-learning-based (Snell et al., 2017; Hofer et al., 2018; Yang and Katiyar, 2020), meta-learning-based (Li et al., 2020a; Sung et al., 2018), prompt-tuning-based (Ma et al., 2022; Hou et al., 2022), and contrastive-learning-based (Das et al., 2022) methods.

However, most existing few-shot NER research has focused on flat entities that do not overlap (Ming et al., 2022; Wang et al., 2022b). In reality, many entities share the same words and form nested entities that are part of another entity. This is where the few-shot nested NER task comes in. This task deals with nested entities that share words

[1]The code is available at https://anonymous.4open.science/r/PACL-840A

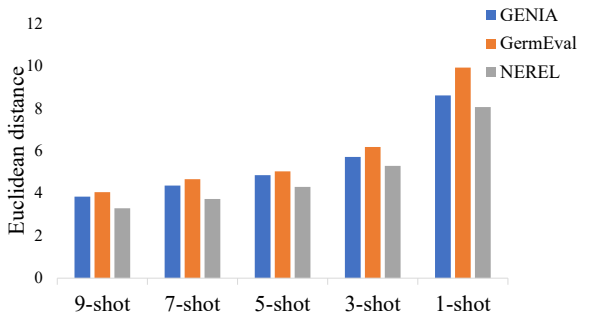


Figure 2: The Euclidean distance of prototype shift between the prototypes in the support set and the query set in the GENIA, GermEval, and NEREL datasets. K-shot denotes the K number of labeled instances in the support set for each type.

and are part of another entity. For example, in the GENIA dataset (Kim et al., 2003), about 53.9% of entities are nested. Figure 1 (a) illustrates an instance, that is, a protein molecule entity "lipoxygenase" is nested within a protein family or group entity "lipoxygenase metabolites". Due to the overlapped part, nested entities are more likely to obtain similar representations, increasing the difficulty of distinguishing them, especially in the few-shot setting where prototype shifts often occur.

The prototype shift in NER refers to changes in the prototypes between the few-shot labeled data set (support set) and unlabeled data (query set), as exemplified in Figure 1 (b), where a prototype is a representative instance of a specific entity type. The very few labeled data in the support set could hardly represent the whole distribution, resulting in prototype shifts. Figure 2 shows the statistics of prototype shifts in terms of Euclidean distance between the support set and the query set in three nested datasets (GENIA (Kim et al., 2003), GermEval (Benikova et al., 2014), NEREL (Loukachevitch et al., 2021)). We can find that the prototype shift reveals a consistent pattern of increasing Euclidean distance between prototypes as the number of labeled data in the support set decreases. When employing the prototypes derived from the support set for delineating the decision boundaries in the query set, a high frequency of classification errors would be introduced due to prototype shifts. Despite having distinguished nested entities within the support set, they may become interspersed within the query set.

This paper addresses the prototype shift in the few-shot nested NER task. Unlike the example-extrapolation-based data augmentation methods (DeVries and Taylor, 2017; Wei, 2021) to enhance the entity representation, we regard entity representation as the combination of prototype and non-prototype representations. Entities of the same type should share the same prototype representation. The non-prototype representation determines the dispersion of the entity distribution. If we could focus more on the prototype representation when learning the entity representation, entities would gather closer around the prototype, and the prototype shift could be reduced. Therefore, we design a prototype-attention mechanism to enhance the prototype representation. Besides, words of the O-type have miscellaneous semantics and cannot be represented by a unique prototype. Therefore, we further design a novel prototype-span contrastive loss. It compares prototype-enhanced span representations with original semantic representations to guarantee the O-type's representations are not enhanced by entity prototypes. It also compares prototype-enhanced span representations with prototypes to reduce prototype differences further.

Our main contributions are as follows:

- We identify the prototype shift challenge in the few-shot learning, particularly in the few-shot nested NER task, and propose a Prototype-Attention Contrastive Learning (PACL) framework to tackle it.
- We devise a unique Prototype-Attention mechanism to generate the prototype-enhanced representation for each span to mitigate the prototype shift between the support and query sets. This mechanism exhibits a high level of generality in enhancing the performance of two baseline models.
- We design a novel prototype-span contrastive loss by comparing prototype-enhanced span representations with prototypes and original semantic representations to reduce prototype differences further and overcome the O-type's non-unique prototype limitation.
- We conduct experiments on various English and cross-lingual nested NER datasets. The results show improvements in PACL over existing nested NER and few-shot NER baselines in terms of F_1 score. Further analyses indicate that our PACL is a simple but effective method and exhibits good generalizability.

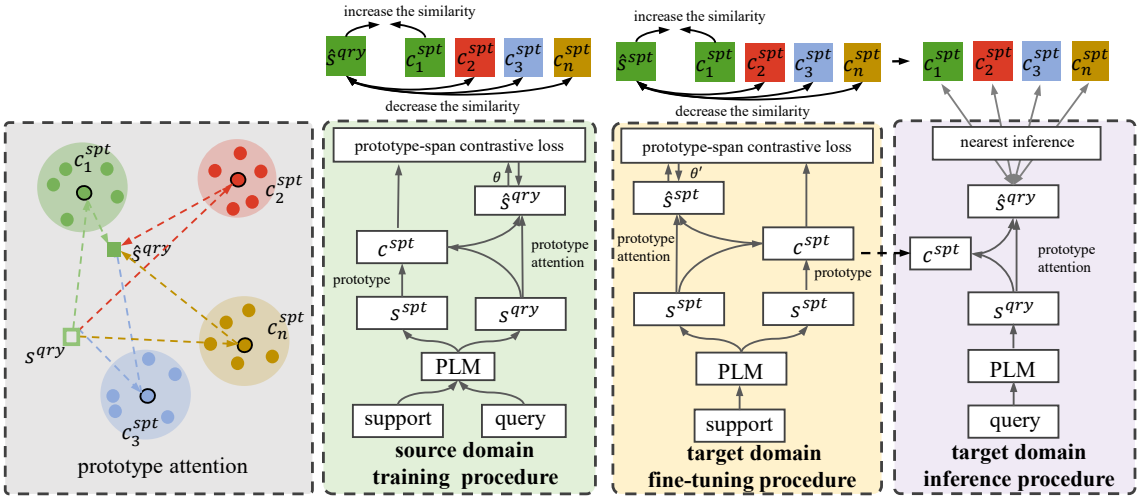


Figure 3: Illustration of our PACL framework and learning procedures. During the training procedure on the source domain, PACL calculates prototypes based on labeled spans of the support set and then utilizes prototype-attention to obtain prototype-enhanced representations for the query set. After that, PACL applies the prototype-span contrastive loss to optimize the representations. During the fine-tuning procedure on the target domain, PACL generates prototype-enhanced representations for the support set to fine-tune the model. Finally, PACL makes inferences on the query set of the target domain based on the nearest neighbor strategy.

2 Problem Definition

Following the mainstream solutions, we formulate the few-shot nested NER task as a span-based entity classification problem. That is, given an input sentence $x \in \mathcal{X}$ with l tokens, denoted by $x = \{w_1, \ldots, w_l\}$, we generate an entity span set containing all possible spans, and each span s_{pq} is a span of tokens starting from the p^{th} token and ending at the q^{th} token in x, denoted by $s_{pq} = \{w_p, \ldots, w_q\}$ $(1 \leqslant p \leqslant q \leqslant l)$. Then, we learn a classification model to map each span into an entity label in the label set $E_{\mathcal{X}}$. If we set the task as a K-shot task, then the number of span labels for each entity type used for training is limited to K. Besides, we also apply the meta-learning framework. The formal descriptions are as follows.

Let $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ denote a dataset with $\mathcal{X}$ and $\mathcal{Y}$ as the sentence set and the corresponding label set, respectively. $\mathcal{D}^{spt} = \{\mathcal{X}^{spt}, \mathcal{Y}^{spt}\}$ and $\mathcal{D}^{qry} = \{\mathcal{X}^{qry}, \mathcal{Y}^{qry}\}$ are disjoint sets sampled from $\mathcal{D}$ for model training and testing, respectively. They are also known as the support set and the query set. Suppose $\mathcal{D}_i = \{\mathcal{X}_i, \mathcal{Y}_i\}$ and $\mathcal{D}_j = \{\mathcal{X}_j, \mathcal{Y}_j\}$ are the source and target domain datasets, respectively. The few-shot nested NER task first samples several subtasks $\{\mathcal{D}_i^{spt}, \mathcal{D}_i^{qry}\}$ from $\mathcal{D}_i = \{\mathcal{X}_i, \mathcal{Y}_i\}$, where $\mathcal{D}_i^{spt} = \{\mathcal{X}_i^{spt}, \mathcal{Y}_i^{spt}\}$, $\mathcal{D}_i^{qry} = \{\mathcal{X}_i^{qry}, \mathcal{Y}_i^{qry}\}$. It then trains a model on these subtasks. After that, it makes adaptations on $\mathcal{D}_j$, i.e., it fine-tunes the model on $\mathcal{D}_j^{spt} = \{\mathcal{X}_j^{spt}, \mathcal{Y}_j^{spt}\}$ and then predicts the span labels for $\mathcal{D}_j^{qry} = \{\mathcal{X}_j^{qry}\}$. For the K-shot setting, each entity category in $\mathcal{X}_i^{spt}$ and $\mathcal{X}_j^{spt}$ contains K entities.

3 Methodology

This section introduces our PACL framework and then provides details of the prototype-attention mechanism, the prototype-span contrastive loss, and target domain adaption procedures.

3.1 PACL Framework

As previously mentioned, the data $\mathcal{D}_i$ in domain i encompasses scenarios where there is a data distribution shift between the support set $\mathcal{D}_i^{spt}$ and the query set $\mathcal{D}_i^{qry}$. We denote the average distribution shift of entity categories as $\mathbb{E}_{\mathcal{D}_i} = \frac{1}{n}\sum_{k=1}^{n}(c_k^{spt} - c_k^{qry})$. c_k^{spt} and c_k^{qry} denote prototype vector for category k in support and query set, respectively. To mitigate the distribution shift, it is necessary to identify a function $\boldsymbol{f}$ that brings the prototype of the query set closer to the support set: $\mathbb{E}_{\mathcal{D}_i} = \frac{1}{n}\sum_{k=1}^{n}(c_k^{spt} - \boldsymbol{f}(c_k^{qry}))$.

Figure 3 illustrates our Prototype-Attention Contrastive Learning (PACL) framework as the function $\boldsymbol{f}$ and learning procedures.

PACL first applies a Pre-trained Language Model (PLM) to obtain the semantic representation for each span. It then calculates prototypes on the

support set and utilizes a novel prototype-attention mechanism to achieve prototype-enhanced representations. After that, PACL optimizes representations by a prototype-span contrastive loss.

During the training procedure on the source domain, PACL utilizes a bunch of subtasks $\{\mathcal{D}_i^{spt}, \mathcal{D}_i^{qry}\}$ to train the model. It generates prototype-enhanced representations for the query set to obtain the adjustment ability for prototype shift. During the fine-tuning procedure on the target domain, PACL utilizes $\mathcal{D}_j^{spt} = \{\mathcal{X}_j^{spt}, \mathcal{Y}_j^{spt}\}$ to fine-tune the model by generating prototype-enhanced representations for the support set. Finally, it predicts the labels for $\mathcal{D}_j^{qry} = \{\mathcal{X}_j^{qry}\}$ by the nearest neighbor strategy.

3.2 Prototype-Attention Mechanism

To mitigate the prototype shift, we propose a prototype-attention (PA) mechanism to generate prototype-enhanced representations for the query set based on prototypes obtained from the support set during training on the source domain. This approach improves the span representations in the query set by incorporating more prototype information, which aligns the prototypes of the query set with those of the support set. The detailed procedures are presented below.

We first incorporate a Pre-trained Language Model (PLM) to obtain original span semantic representations. For the sentence x with l tokens, we get all word embeddings, concatenate the start and the end token embeddings of each span, and use a non-linear function to get the span semantic representation $\boldsymbol{s}$:

$$[\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_l}] = \mathrm{PLM}([\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_l]) \quad (1)$$

$$\boldsymbol{s} = ReLU(\boldsymbol{h_p} \oplus \boldsymbol{h_q}) \quad (2)$$

Where $\oplus$ denotes the concatenation operator.

Prototype based methods predict the probabilities of each $\boldsymbol{s}$ as Equation 3

$$\boldsymbol{P}(\hat{y}_k) = \frac{exp(-d(\boldsymbol{s}, \boldsymbol{c}_k))}{\sum_{k \in K} exp(-d(\boldsymbol{s}, \boldsymbol{c}_k))} \quad (3)$$

Where $\boldsymbol{c}_k$ is the prototype of entity type k and calculated via mean-pooling of each span representation in type k. And $d(\cdot)$ is the normalized cosine-similarity. The final predicted label for the span $\boldsymbol{s}$ is given by

$$\hat{y} = argmax_{k \in \{1,2,3\ldots\}} \boldsymbol{P}(\hat{y}_k) \quad (4)$$

To mitigate prototype shift for entity spans in the support set and the query set, we gain the prototype-enhanced representation $\hat{\boldsymbol{s}}^{qry}$ in the query set by calculating the attention score between the original span representation $\boldsymbol{s}^{qry}$ and prototypes $\mathcal{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots]$ in the support set:

$$\hat{\boldsymbol{s}}^{qry} = softmax\left(\frac{\boldsymbol{s}^{qry}\,\mathcal{C}^\top}{\sqrt{d_\mathcal{C}}}\right)\mathcal{C} + \boldsymbol{s}^{qry} \quad (5)$$

where $d_\mathcal{C}$ is the dimension of prototypes. We also include $\boldsymbol{s}^{qry}$ in the attention representation to obtain $\hat{\boldsymbol{s}}^{qry}$, excluding the O-type spans which cannot be represented by prototypes in $\mathcal{C}$. This will be further optimized in the next section with prototype-span contrastive loss.

3.3 Prototype-Span Contrastive Loss

The traditional contrastive loss increases span similarities of the same entity type and decreases span similarities between different entity types. This paper aims to address the prototype shift. Therefore, we want to increase the similarity between spans in the query set and the corresponding prototype in the support set to let the model obtain the ability to mitigate the prototype shift. Besides, the O-type span has miscellaneous semantics and could not be represented by a unique prototype (Fritzler et al., 2019). We also want prototype-enhanced representations of O-type entities close to their original semantic representations. Therefore, we design the following prototype-span contrastive loss based on the circle loss (Sun et al., 2020).

For each span representation $\hat{\boldsymbol{s}}^{qry}$ in the query set, the loss $\mathcal{L}_{\hat{\boldsymbol{s}}^{qry}}$ is calculated by:

$$\mathcal{L}_{\hat{\boldsymbol{s}}^{qry}} = log(1 + sim(\hat{\boldsymbol{s}}^{qry}, \boldsymbol{c}^+) * sim(\hat{\boldsymbol{s}}^{qry}, \boldsymbol{c}^-)) \quad (6)$$

Where $\boldsymbol{c}^+$ is the corresponding prototype in the support set with the same type as $\hat{\boldsymbol{s}}^{qry}$, and $\boldsymbol{c}^-$ denotes prototypes in the support set with different types from $\hat{\boldsymbol{s}}^{qry}$. The similarity function sim is calculated by:

$$sim(\hat{\boldsymbol{s}}^{qry}, \boldsymbol{c}^+) = e^{-\tau * \phi(\hat{\boldsymbol{s}}^{qry}, \boldsymbol{c}^+)} \quad (7)$$

$$sim(\hat{\boldsymbol{s}}^{qry}, \boldsymbol{c}^-) = \sum_{\boldsymbol{c}_i^- \in \boldsymbol{c}^-} e^{\tau * \phi(\hat{\boldsymbol{s}}_i^{qry}, \boldsymbol{c}_i^-)} \quad (8)$$

Where $\phi(.)$ denotes the cosine similarity, τ is the temperature (Wang and Liu, 2021).

When calculating $sim(\hat{\boldsymbol{s}}^{qry}, \boldsymbol{c}^+)$ for the O-type, we calculate the cosine similarity between the orig-

inal span representation $\boldsymbol{s}^{qry}$ and the prototype-enhanced representation $\hat{\boldsymbol{s}}^{qry}$:

$$\phi(\hat{\boldsymbol{s}}_i^{qry},\ \boldsymbol{c}_o) = \lambda * \phi(\hat{\boldsymbol{s}}_i^{qry}, \boldsymbol{s}^{qry}) \quad (9)$$

Where λ is a learnable hyperparameter. We calculate the cosine similarity between the prototype-enhanced representation $\hat{\boldsymbol{s}}^{qry}$ and its corresponding prototype in the support set for other entity types.

3.4 Target Domain Adaption

After training the model on the source domain, we make adaptions to the target domain, including fine-tuning the model on the support set and making inferences on the query set.

During the fine-tuning procedure, our PACL first generates prototype-enhanced representations $\hat{\boldsymbol{s}}^{spt}$ for spans in the support set by calculating the attention score between the original span representation $\boldsymbol{s}^{spt}$ and the prototypes $\mathcal{C}$ in the support set. After that, PACL fine-tunes the model by utilizing the prototype-span contrastive loss with the input of $\hat{\boldsymbol{s}}^{spt}$ and $\mathcal{C}$. Different from using $\hat{\boldsymbol{s}}^{qry}$ as the input in the training procedure, we utilize $\hat{\boldsymbol{s}}^{spt}$ in the fine-tuning procedure since the labels of the query set are unknown.

During the inference procedure, our PACL obtains prototype-enhanced representations $\hat{\boldsymbol{s}}^{qry}$ for spans in the query set according to prototypes $\mathcal{C}$ in the support set. It further applies the nearest neighbor inference for each span according to the maximum similarity with prototypes or its original span representation (O-type).

4 Experiments

In this section, we evaluate PACL in few-shot nested NER. After introducing datasets and baseline models, we outline the setup, present results, and analyze them thoroughly.

4.1 Datasets

To evaluate our proposed PACL, We validate the effectiveness of our model on English datasets and assess its robustness and generalizability in cross-lingual setting by German, Russian, and Chinese datasets.

As shown in Table 1, the English target nested NER datasets are ACE04[2] (Doddington et al., 2004), ACE05[3] (Ntroduction), and GENIA[4]. (Kim et al., 2003). And the cross-lingual target nested NER datasets are Chinese part of ACE05, GermEval[5] in German (Benikova et al., 2014), and NEREL[6] in Russian (Loukachevitch et al., 2021). To ensure the complete difference with the target test domain, We use a flat NER dataset, FewNERD [7] in English (Ding et al., 2021), as the source domain dataset to train the model. We have manually checked to guarantee these datasets are without offensive content and identifiers.

Dataset	language	Types	Sentences	Entities/Nest entities
ACE04	English	7	6.8k	27.8k / 12.7k
ACE05	English	7	13.6k	50.2k / 18.3k
ACE05_Chinese	Chinese	7	6.5k	34.2k / 15.5k
GENIA	English	36	18.5k	55.7k / 30.0k
GermEval	German	12	18.4k	41.1k / 6.1k
NEREL	Russian	29	8.9k	56.1k / 18.7k
FewNERD	English	66	188.2k	491.7k / -

Table 1: Datasets used in experiments

For training in the source domain, We randomly sampled 10,500 5-way 5-shot subtasks from the FewNERD inter-domain subset, among which 10,000 subtasks are used for training and 500 subtasks are used for validation. We validated the model every 1000 subtasks. When fine-tuning in the target domain, we sampled 32-way support sets under 1-shot and 5-shot settings from the GENIA dataset instead of all 36-way due to there being 4 types having less than 50 entities. For other datasets, we sample all types to make the few-shot support set.

4.2 Baselines

We compare our proposed PACL with seven baselines which can be categorized into three groups: 1) Rich-resource nested NER methods including NER-DP (Yu et al., 2020), IoBP (Wang et al., 2021), and PO-TreeCRFs (Fu et al., 2021); 2) Metric-based few-shot NER methods including ProtoNet (Snell et al., 2017), NNShot (Yang and Katiyar, 2020), ESD (Wang et al., 2022c), and SpanProto (Wang et al., 2022a); 3) Contrastive-learning-based few-shot NER method CONTaiNER (Das et al., 2021). Appendix A details these baseline models.

[2] https://catalog.ldc.upenn.edu/LDC2005T09
[3] https://catalog.ldc.upenn.edu/LDC2006T06
[4] http://www.geniaproject.org/genia-corpus
[5] https://sites.google.com/site/germeval2014ner/data
[6] https://github.com/nerel-ds/NEREL
[7] https://ningding97.github.io/fewnerd/

Model	ACE04 (7-way)		ACE05 (7-way)		GENIA (32-way)		Average	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
NER-DP	4.01±2.75	11.48±4.05	6.48±5.34	15.58±8.54	14.26±3.98	32.19±4.59	8.25	19.75
IoBP	10.63±6.70	14.14±6.06	15.68±4.48	34.36±6.62	15.14±2.34	19.89±5.47	13.82	22.80
PO-TreeCRFs	10.55±4.79	29.77±7.97	18.02±11.93	33.83±10.54	21.88±4.32	40.21±3.67	16.82	34.60
CONTaiNER	6.87±2.89	14.19±3.09	11.46±3.30	15.52±4.96	8.39±1.33	10.92±1.76	8.91	13.54
ProtoNet	25.55±8.23	40.18±6.19	25.61±11.25	41.52±5.14	20.52±3.86	36.02±3.28	23.89	39.24
NNShot	22.01±7.92	37.74±5.55	23.93±10.74	36.69±6.23	23.87±3.79	36.01±2.33	23.27	36.81
ESD	23.41±6.19	39.13±5.09	24.85±11.17	41.30±5.37	21.11±4.15	26.79±1.77	23.12	35.74
SpanProto	24.90±5.80	40.10±5.98	**29.92**±8.27	41.65±7.89	30.91±2.74	40.95±1.52	28.33	40.90
PACL	**30.31**±6.15	**43.16**±6.86	29.35±10.08	**46.47**±6.38	**33.89**±2.15	**44.76**±1.61	**31.18**	**44.80**

Table 2: F_1 performance on English datasets (%).

Model	ACE05_Chinese (7-way)		GermEval (12-way)		NEREL (29-way)		Average	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
NER-DP	5.06±2.60	15.82±8.33	6.99±3.05	20.25±5.14	14.18±3.78	40.87±3.96	8.74	25.65
IoBP	-	-	3.27±2.17	20.99±13.19	8.36±1.27	23.66±2.91	5.82	22.33
PO-TreeCRFs	7.94±5.43	15.43±4.64	6.60±5.63	40.06±5.17	17.91±3.21	44.48±2.98	10.82	33.32
CONTaiNER	12.27±3.70	16.09±5.45	12.38±2.81	17.81±3.23	14.84±2.38	27.09±2.15	13.16	20.33
ProtoNet	36.36±6.70	51.40±2.52	35.75±6.57	49.68±4.16	41.83±3.83	56.50±2.09	37.98	52.53
NNShot	36.47±5.17	51.73±4.54	36.51±7.30	46.07±10.60	41.53±2.81	57.99±2.52	38.17	51.93
ESD	33.95±6.76	47.64±3.14	34.13±8.15	35.29±5.71	34.86±4.12	46.30±4.70	34.31	43.08
SpanProto	37.41±4.97	51.16±4.55	38.30±7.52	50.10±2.97	44.09±3.62	57.47±1.95	39.93	52.91
PACL	**44.65**±5.93	**54.90**±3.58	**47.53**±6.30	**58.43**±2.69	**50.26**±4.60	**62.08**±1.36	**47.48**	**58.47**

Table 3: Micro F_1 performance on cross-lingual datasets (%).

4.3 Experimental Settings

We implemented PACL by Huggingface Transformer 4.21.1 and PyTorch 2.1. The model is initialized randomly and optimized by AdamW (Loshchilov and Hutter, 2017). We train and fine-tune the model with the learning rate 5e-5. For the text encoder, we use the pre-trained $\text{BERT}_{\text{base\_multilingual}}$ model since the languages of target domain datasets are different. The hidden layer of the non-linear function f in equation 2 for getting span semantic representations is set to 512, and the initial value of the learnable hyperparameter λ for the O-type is set to 0.5. We set random seeds ranging from 0 to 10 to get ten results for each setting and report the average and standard deviation values to evaluate all models.

4.4 Experimental Results

To evaluate the effectiveness of our PACL, we compare it against state-of-the-art baseline models introduced in 4.2. Table 2 and Table 3 shows their average F_1 results on English test datasets and cross-lingual test datasets.

For the English test sets, except for the ACE05-1shot setting, our proposed PACL model surpasses the baseline model. Specifically, under the 1-shot setting, our method outperforms the baseline method by 5.41% and 2.98% on the ACE04 and GENIA datasets, respectively. Under the 5-shot setting, our method surpasses the baseline method by 3.06%, 4.82%, and 3.81% on the ACE04, ACE05, and GENIA datasets, respectively.

And for the cross-lingual test sets, except for the IoBP's inability to handle Chinese tasks (indicated as "-" in the table 3), our proposed PACL model consistently outperforms the baseline model. Specifically, for the ACE05_Chinese, GermEval, and NEREL datasets, under the 1-shot setting our method surpasses the baseline method by 7.24%, 9.23%, and 6.17%, respectively. And under the 5-shot setting, our method outperforms the baseline method by 3.74%, 8.33%, and 4.61%, respectively.

Overall, these results demonstrate the effectiveness of our proposed PACL framework compared to the state-of-the-art baseline models.

4.5 Experimental Analysis

This section presents ablation studies, results on only-nested in the test datasets, the generality of the PA mechanism, and the efficiency Study of the PA mechanism.

4.5.1 Ablation Study

To evaluate the contribution of the designed PA mechanism to the overall performance of PACL, we conduct the ablation study by removing PA from the PACL. The detailed results shown in Appendix D suggest that the PA mechanism positively impacts the F_1 score for both the English and the

corss-lingual tasks. On average, the PA mechanism improves performance on both English and cross-lingual tasks by 4.04% and 2.91%, respectively, under the 1-shot setting. And under the 5-shot setting, it improves performance by 1.65% and 3.56%, respectively.

In conclusion, the PA module has an overall positive impact, primarily because it reduces the prototype shift. Appendix B shows how our PACL mitigates the prototype shift.

4.5.2 Only-Nested Results

In order to more comprehensively demonstrate the efficacy of the outcomes pertaining to nested entities across these datasets, we undertook a process of splitting and filtering exclusively for nested entities. The results in Appendix C show that for the English test dataset, except for the GENIA-1shot setting, our proposed PACL model can also outperform the baseline models in predicting only nested entities. And for cross-lingual datasets, our PACL model consistently outperforms the baseline model.

Specifically, for the English test datasets, only under the GENIA-1shot setting did PACL fail to surpass the baseline model. Apart from this, for ACE04 and ACE05 under the 1-shot setting, our PACL outperforms the baseline model by 4.54% and 2.57% respectively. Under the 5-shot condition, for ACE04, ACE05, and the GENIA dataset, our PACL surpasses the baseline model by 4.79%, 3.99%, and 1.22%. For cross-language test datasets, PACL outperforms all baseline models. Under the 1-shot setting for ACE05_Chinese, GermEval, and NEREL, our PACL surpasses the baseline models by 4.62%, 5.07%, and 2.62%, respectively. Meanwhile, under the 5-shot setting for ACE05_Chinese, GermEval, and NEREL datasets, our PACL outperforms the baseline models by 1.29%, 6.73%, and 1.89%, respectively.

In summary, our proposed PACL demonstrates advantages over the baseline models in identifying nested entities.

4.5.3 Generality of Prototype-Attention Mechanism

As the Prototype-Attention (PA) mechanism addresses the fundamental property of the prototype shift phenomenon, we believe it has a high level of generalizability and can enhance the performance of various models.

To assess the generality of the PA mechanism, we conduct experiments by integrating it into the SpanProto and ESD models and comparing the performance before and after integration. As shown in Appendix E, the experiment results demonstrate that integrating the PA mechanism into SpanProto and ESD improves the F_1 score on several datasets.

These findings suggest that the PA mechanism has high generality and can serve as a valuable tool for NLP practitioners looking to improve their models' performance in few-shot nested NER tasks.

4.5.4 Efficiency Study

Through our time analysis in Appendix F, we found that our proposed PACL spends a similar amount of time as ESD during fine-tuning on the few-shot support set, but it requires less time than baseline models during testing on the unlabeled query set. Additionally, using the PA mechanism incurs only extremely minor overhead. As demonstrated in section 4.5.1 and 4.5.3, PA mechanism can enhance the predictive performance of the models. This indicates that the PA method is simple but effective.

4.5.5 Case Study

Figure 5 in Appendix G displays instances from several datasets along with the prediction results of our proposed PACL model. Currently, the results of the few-shot nested NER tasks are not particularly satisfactory. Our proposed PACL method may also suffer from missed recognitions and identification errors, but overall, it exhibits good performance.

5 Related Work

This section discusses related works on rich-resource nested NER, few-shot NER, and distribution shifts.

5.1 Rich-resource Nested NER

Nested NER aims to recognize entities with nested structures. Most of the current methods for nested NER are established on rich-resource datasets. These methods could be categorized into span-based, hypergraph-based, and layered-based (Wan et al., 2022).

Span-based methods treat sequences of tokens as spans and then label all possible spans by classification models (Shen et al., 2021; Li et al., 2020b; Tan et al., 2021). Hypergraph-based methods analyze the dependence of words in a sentence and then construct a dependency tree (Yu et al., 2020) or other structures (Wang and Lu, 2018; Katiyar and Cardie, 2018) to help identify nested entities. And layered-based methods capture the depth of

entity nesting and apply multi-level sequence labeling strategies to recognize nested entities (Wang et al., 2021; Shibuya and Hovy, 2020).

These methods may be stuck in overfitting due to sophisticated models and the limited number of instances for training in the few-shot setting.

5.2 Few-shot NER

Few-shot NER requires recognizing entities with the support of very few labeled instances (Hofer et al., 2018; Fritzler et al., 2019). Due to limited information contained in the support set, methods for few-shot NER mainly resort to a rich-resource source domain to help train models, resulting in transfer-learning and meta-learning frameworks.

Transfer-learning-based methods train models on a source domain and then transfer models or features to the few-labeled target domain (Yang et al., 2021; Liu et al., 2021). Meta-learning-based methods train models on adequate subtasks to make the model acquire the learning ability on few-shot tasks (de Lichy et al., 2021; Li et al., 2020a). Comparatively speaking, meta-learning-based methods are more widely used in few-shot NER due to their easy adaption to new tasks.

Within the meta-learning framework, various kinds of models are designed. For example, metric-based methods, including ProtoNet (Snell et al., 2017), NNShot (Yang and Katiyar, 2020), and SpanProto (Wang et al., 2022a), measure distances between prototypes in the support set and instances in the query set. Optimization-based methods, such as MAML (Finn et al., 2017) and FEWNER (Li et al., 2020a), train the model by a special optimizer. Model-based methods, such as SNAIL (Mishra et al., 2017) and CNPs (Garnelo et al., 2018), learn the hidden representation of instances on the support set and the query set to make inferences in an end-to-end manner. Contrastive-learning methods, such as CONTaiNER (Das et al., 2022), aims to maximize similarities of the same type and minimize similarities between different types.

These few-shot NER methods mostly focus on flat entities. Few works have discussed the few-shot nested NER setting. Wang converted sequence labeling to span-level matching for the few-shot flat NER and showed their method could handle nested entities (Wang et al., 2022b). However, it is not designed for the few-shot nested NER specifically.

5.3 Distribution Shifts

Distribution shift is a problem of training and testing data following two different distributions. It affects the generalization ability of supervised deep-learning models as the fundamental that these models could work is that training and testing data come from the same distribution. Inspired by real-world challenges, Wiles et al. summarized three distribution shifts: spurious correlation, low-data drift, and unseen data shift (Wiles et al., 2022). There have been some researches aiming to address distribution shifts in computer vision and general natural language processing tasks (Fang et al., 2020; Tu et al., 2022). To the best of our knowledge, researchers seldom discuss the distribution shift problem in the few-shot NER task. In this paper, we aim to tackle the few-shot nested NER task. Therefore, we rethink the distribution shift problem from the perspective of entity representation distribution and identify the prototype shift since it directly affects entity classification.

6 Conclusion

This paper first identifies the phenomenon of prototype shift that arises when there is a difference in prototypes between the support and query sets. Within the context of few-shot learning tasks, prototype shift is prone to occur since the few labeled instances in the support set could hardly represent the query set. To mitigate this issue in the few-shot nested NER task, we propose the Prototype-Attention Contrastive Learning (PACL) framework combining a prototype-attention mechanism and a prototype-span contrastive loss to enhance prototype representations. The experiments on English tasks show the effectiveness of PACL and the experiments on cross-lingual tasks show the robustness of PACL. Furthermore, our prototype-attention mechanism applied to baseline models also leads to performance improvements, further validating the strong generalizability of our approach.

7 Limitations

This paper still has several limitations. The first one is about the prototype shift adjustment. It is hard to completely address the prototype shift, while our PACL makes this attempt and achieves inspiring improvement. The second one is about other distribution shifts. Prototype shift is just one kind of distribution shift. Other distribution shifts also need

to be identified and addressed to improve the accuracy of the few-shot nested NER task. The third one is about the language used for training. We utilized FewNERD as the source domain training dataset and conducted testing tasks on the English datasets including ACE04 and ACE05. These two datasets belong to the MIX domain, encompassing various types of entities. Although FewNERD is a flat dataset while ACE04/05 are nested datasets, there exists a potential risk of training domain information leaking into the target domain.

References

Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. Nosta-d named entity annotation for german: Guidelines and dataset. In *LREC*, pages 2524–2531.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J Passonneau, and Rui Zhang. 2021. Container: Few-shot named entity recognition via contrastive learning. *arXiv preprint arXiv:2109.07589*.

Cyprien de Lichy, Hadrien Glaude, and William Campbell. 2021. Meta-learning for few-shot named entity recognition. In *Proceedings of the 1st Workshop on Meta Learning and Its Applications to Natural Language Processing*, pages 44–58.

Terrance DeVries and Graham W Taylor. 2017. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. 2020. Rethinking importance weighting for deep learning under distribution shift. *Advances in Neural Information Processing Systems*, 33:11996–12007.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Omar U Florez and Erik Mueller. 2019. Learning to control latent representations for few-shot learning of named entities. *arXiv preprint arXiv:1911.08542*.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 993–1000.

Yao Fu, Chuanqi Tan, Mosha Chen, Songfang Huang, and Fei Huang. 2021. Nested named entity recognition with partially-observed treecrfs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12839–12847.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. 2018. Conditional neural processes. In *International Conference on Machine Learning*, pages 1704–1713. PMLR.

Maximilian Hofer, Andrey Kormilitzin, Paul Goldberg, and Alejo Nevado-Holgado. 2018. Few-shot learning for named entity recognition in medical text. *arXiv preprint arXiv:1811.05468*.

Yutai Hou, Cheng Chen, Xianzhen Luo, Bohan Li, and Wanxiang Che. 2022. Inverse is better! fast and accurate prompt for few-shot slot tagging. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 637–647, Dublin, Ireland. Association for Computational Linguistics.

Yutai Hou, Zhihan Zhou, Yijia Liu, Ning Wang, Wanxiang Che, Han Liu, and Ting Liu. 2019. Few-shot sequence labeling with label dependency transfer and pair-wise embedding. *arXiv preprint arXiv:1906.08711*.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Jing Li, Billy Chiu, Shanshan Feng, and Hao Wang. 2020a. Few-shot named entity recognition via meta-learning. *IEEE Transactions on Knowledge and Data Engineering*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020b. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Natalia Loukachevitch, Ekaterina Artemova, Tatiana Batura, Pavel Braslavski, Ilia Denisov, Vladimir Ivanov, Suresh Manandhar, Alexander Pugachev, and Elena Tutubalina. 2021. NEREL: A Russian dataset with nested named entities, relations and events. In *Proceedings of RANLP*, pages 876–885.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022. Template-free prompt tuning for few-shot NER. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732, Seattle, United States. Association for Computational Linguistics.

Hong Ming, Jiaoyun Yang, Lili Jiang, Yan Pan, and Ning An. 2022. Few-shot nested named entity recognition. *arXiv preprint arXiv:2212.00953*.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2017. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*.

Ii. I Ntroduction. The ace 2005 (ace 05) evaluation plan evaluation of the detection and recognition of ace entities , values , temporal expressions , relations , and events 1.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794, Online. Association for Computational Linguistics.

Takashi Shibuya and Eduard Hovy. 2020. Nested Named Entity Recognition via Second-best Sequence Learning and Decoding. *Transactions of the Association for Computational Linguistics*, 8:605–620.

Jake Snell, Kevin Swersky, and Richard S Zemel. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*.

Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. 2020. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021. A sequence-to-set network for nested named entity recognition. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3936–3942. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Jianhong Tu, Ju Fan, Nan Tang, Peng Wang, Chengliang Chai, Guoliang Li, Ruixue Fan, and Xiaoyong Du. 2022. Domain adaptation for deep entity resolution. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, pages 443–457. ACM.

Juncheng Wan, Dongyu Ru, Weinan Zhang, and Yong Yu. 2022. Nested named entity recognition with span-level graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–903, Dublin, Ireland. Association for Computational Linguistics.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 204–214, Brussels, Belgium. Association for Computational Linguistics.

Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504.

Jianing Wang, Chengyu Wang, Chuanqi Tan, Minghui Qiu, Songfang Huang, Jun Huang, and Ming Gao. 2022a. Spanproto: A two-stage span-based prototypical network for few-shot named entity recognition. *CoRR*, abs/2210.09049.

Peiyi Wang, Runxin Xu, Tianyu Liu, Qingyu Zhou, Yunbo Cao, Baobao Chang, and Zhifang Sui. 2022b. An enhanced span-based decomposition method for few-shot sequence labeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5012–5024, Seattle, United States. Association for Computational Linguistics.

Peiyi Wang, Runxin Xu, Tianyu Liu, Qingyu Zhou, Yunbo Cao, Baobao Chang, and Zhifang Sui. 2022c. An enhanced span-based decomposition method for few-shot sequence labeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5012–5024, Seattle, United States. Association for Computational Linguistics.

Yiran Wang, Hiroyuki Shindo, Yuji Matsumoto, and Taro Watanabe. 2021. Nested named entity recognition via explicitly excluding the influence of the best path. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3547–3557, Online. Association for Computational Linguistics.

Jason Wei. 2021. Good-enough example extrapolation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5923–5929, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dj Dvijotham, and Ali Taylan Cemgil. 2022. A fine-grained analysis on distribution shift. In *International Conference on Learning Representations*.

Guanqun Yang, Shay Dineen, Zhipeng Lin, and Xueqing Liu. 2021. Few-sample named entity recognition for security vulnerability reports by fine-tuning pre-trained language models. In *International Workshop on Deployable Machine Learning for Security Defense*, pages 55–78. Springer.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. *arXiv preprint arXiv:2010.02405*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

A Detail of Baselines

Detailed information on baseline models is introduced in this section. We compare our PACL with the following seven baseline models:

- NER-DP (Yu et al., 2020) is a rich-resource-based nested NER method. It applies a biaffine model to score pairs of start and end tokens for each span to establish dependency parsing for identifying nested entities.
- IoBP (Wang et al., 2021) introduces a significant enhancement to NER, leveraging the second-best path recognition method's framework while reducing the impact of the best path. This approach adopts a layered architecture, preserving a set of hidden states at each temporal iteration. These states are subsequently employed to construct diverse potential functions for recognizing nested entities across various hierarchical levels.
- PO-TreeCRFs (Fu et al., 2021) tackles the challenge of nested NER by conceptualizing it as a constituency parsing issue with partially observed trees. Introducing a fresh model called partially observed TreeCRFs, this approach regards labeled entity spans as observed nodes within a constituency tree, while the remaining spans are considered latent nodes.
- CONTaiNER (Das et al., 2021) is a contrastive-learning-based few-shot NER method. It first obtains entities' Gaussian-distributed embeddings and then optimizes a generalized objective of differentiating between entity types by a contrastive loss function. We adapt it to handle nested entities with the entity span formulation.
- ProtoNet (Snell et al., 2017) is a metric-learning-based few-shot NER method. It applies prototypical networks to learn a metric space for obtaining prototype representations. We also adapt it to handle nested entities with the entity span formulation.
- NNShot (Yang and Katiyar, 2020) is also a metric-learning-based few-shot NER method. It applies structured decoding and nearest-neighbor learning to identify entities. We utilize the entity span formulation to make it handle nested entities.
- ESD (Wang et al., 2022c) is a metric-learning-based few-shot NER method. It formulates the task as a span-level matching problem. To identify entities, it performs span-level procedures, including enhanced span representation, class prototype aggregation, and span conflict resolution.
- SpanProto (Wang et al., 2022a) is a metric-learning-based few-shot NER method. It also

model	ACE04 (7-way)		ACE05 (7-way)		GENIA (32-way)		Average	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
NER-DP	3.44±2.28	9.49±2.52	3.61±2.56	9.63±3.88	13.54±3.76	30.01±3.83	6.86	16.37
CONTaiNER	4.66±2.09	10.01±1.45	4.73±0.83	8.70±2.68	7.33±1.53	9.46±1.65	5.57	9.39
ProtoNet	19.17±6.22	31.44±3.66	16.94±6.90	30.10±3.18	17.76±3.86	32.07±2.60	17.96	31.20
NNShot	16.70±6.65	30.59±3.60	15.70±6.98	27.07±3.06	23.77±3.77	34.70±1.85	18.72	30.79
ESD	16.18±5.09	29.10±4.41	15.39±6.48	28.11±3.68	18.55±4.90	23.86±2.78	16.71	27.02
SpanProto	18.19±5.38	30.47±4.94	19.02±4.96	30.50±4.61	**31.43**±3.17	39.89±1.38	22.88	33.62
PACL	**22.73**±6.46	**35.26**±5.01	**21.59**±7.43	**34.49**±2.65	30.51±2.09	**41.11**±1.81	**24.94**	**36.95**

Table 4: Micro F_1 performance on English datasets with only-nested entity setting (%).

model	ACE05_Chinese (7-way)		GermEval (12-way)		NEREL (29-way)		Average	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
NER-DP	4.87±2.10	13.62±7.29	6.27±2.72	14.08±4.13	7.05±3.67	24.86±4.31	6.06	17.52
CONTaiNER	9.27±2.82	12.47±4.61	10.41±3.45	10.63±2.96	6.64±2.10	7.78±1.44	8.77	10.29
ProtoNet	32.66±5.57	38.50±2.39	19.85±5.54	31.66±5.79	25.72±4.88	39.83±2.18	26.08	36.66
NNShot	31.55±4.96	46.05±4.57	26.96±6.88	29.18±7.38	27.61±4.22	43.21±4.09	28.71	30.46
ESD	28.69±5.43	38.26±3.10	22.39±3.91	22.97±3.67	20.71±3.98	30.16±4.96	23.93	30.46
SpanProto	35.10±4.66	45.83±4.23	25.19±5.65	34.00±5.91	30.66±4.25	44.06±2.33	30.32	41.30
PACL	**39.72**±4.75	**47.12**±4.21	**30.26**±8.12	**40.73**±2.00	**33.28**±5.91	**45.95**±2.14	**34.42**	**44.60**

Table 5: Micro F_1 performance cross-lingual datasets with only-nested entity setting (%).

applies entity spans to formulate the problem. For identifying entities, it first utilizes a span extractor to recognize candidate entity spans and then applies a mention classifier to determine entity types.

B Prototype Shift Mitigation by PACL

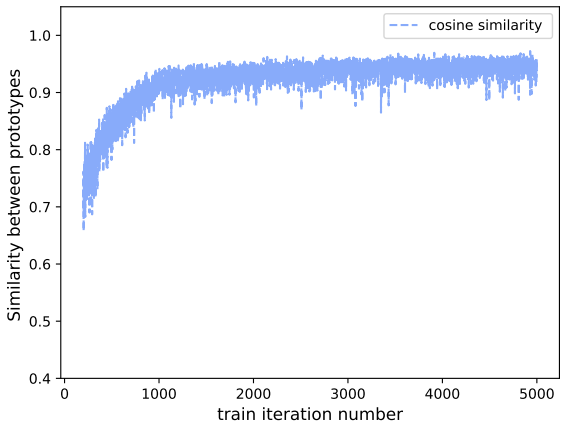


Figure 4: Illustration of the change of the prototype similarity during training.

This paper aims to mitigate prototype shifts, and section 1 has already validated the existence of the prototype shift phenomenon. This section examines how the prototype shift changes by applying our PACL.

We utilize the cosine similarity to denote the prototype differences between the support and query sets to measure the prototype shift. Figure 4 illustrates the change of the prototype similarity with the increase of iteration numbers during training. We could find a consistently increasing trend in prototype similarity, which means the prototype shift is consistently decreasing. This validates the effectiveness of our PACL in mitigating prototype shifts.

C Results on Only-Nested Entities

We only calculate the F_1 value of nested entity recognition for unlabeled query sets. Table 4 and Table 5 show the average F_1 results of only-nested entities on English and cross-lingual test datasets with 1-shot and 5-shot settings. For the English test dataset, we observe that except for the GENIA-1shot setting, our proposed PACL model can also outperform the baseline model in recognizing only nested entities. As for cross-lingual datasets, our PACL model consistently outperforms the baseline model in all settings.

Our proposed model achieves the best results in almost all experimental settings and datasets, indicating that our model has an advantage in recognizing nested entities compared to other baseline models.

D Ablation Study

Table 7 presents the ablation experiments for the PA module. Except for the ACE04-5shot setting, removing the PA module results in a decrease in model performance. Specifically, for the ACE04, ACE05, GENIA, ACE05_Chinses, GermEval, and NEREL datasets, under the 1-shot experimental set-

		SpanProto	SpanProto w PA	ESD	ESD w PA
ACE04	1-shot	24.90±5.80	27.32±6.34 ↑	23.41±6.19	22.63±7.61 ↓
	5-shot	40.10±5.98	44.67±4.29 ↑	39.13±5.09	38.22±5.20 ↓
ACE05	1-shot	29.92±8.27	29.56±11.24 ↑	24.85±11.17	23.80±8.95 ↓
	5-shot	41.65±7.89	45.40±5.45 ↑	41.30±5.37	41.02±7.30 ↓
GENIA	1-shot	30.91±2.74	30.95±2.23 ↑	21.11±4.15	30.69±3.30 ↑
	5-shot	40.95±1.52	42.23±1.98 ↑	26.79±1.77	40.49±3.51 ↑
ACE05_Chinese	1-shot	37.41±4.97	42.81±6.66 ↑	33.95±6.76	35.93±6.86 ↑
	5-shot	51.16±4.55	54.31±3.90 ↑	47.64±3.14	49.56±2.94 ↑
GermEval	1-shot	38.30±7.52	45.48±4.24 ↑	34.13±8.15	36.00±6.68 ↑
	5-shot	50.10±2.97	56.06±2.55 ↑	35.29±5.71	49.94±3.24 ↑
NEREL	1-shot	44.09±3.62	49.72±3.97 ↑	34.86±4.12	42.14±3.72 ↑
	5-shot	57.47±1.95	61.43±1.24 ↑	46.30±4.70	58.08±1.24 ↑

Table 6: F_1 performance before and after integrating the Prototype-Attention (PA) mechanism to SpanProto and ESD on test datasets (%).

		PACL	w/o PA
ACE04	1-shot	30.31±6.15	26.68±6.29 ↓
	5-shot	43.16±6.86	44.07±5.53 ↑
ACE05	1-shot	29.35±10.08	26.41±9.62 ↓
	5-shot	46.47±6.38	43.68±6.58 ↓
GENIA	1-shot	33.89±2.15	28.35±2.94 ↓
	5-shot	44.76±1.61	41.70±1.66 ↓
ACE05_Chinese	1-shot	44.65±5.93	41.35±6.54 ↓
	5-shot	54.90±3.58	53.33±3.84 ↓
GermEval	1-shot	47.53±6.30	46.02±6.16 ↓
	5-shot	58.43±2.69	52.49±3.87 ↓
NEREL	1-shot	50.26±4.60	46.35±2.74 ↓
	5-shot	62.08±1.36	58.92±1.82 ↓

Table 7: Ablation study of F_1 performance on test datasets (%). "**w/o PA**" means removing the Prototype-Attention mechanism.

ting, removing the PA module leads to a decrease in the model's final F_1 score by 3.63%, 2.94%, 5.54%, 3.30%, 1.51%, and 3.91%, respectively. Furthermore, under the 5-shot experimental setting, removing the PA module results in a decrease in the model's final F_1 score by 2.79%, 3.06%, 1.57%, 5.94%, and 3.16% for the ACE05, GENIA, ACE05_Chinese, GermEval, and NEREL datasets, respectively.

E Generality of Prototype-Attention Mechanism

We applied the PA mechanism to the SpanProto and ESD. The results are shown in Table 6.

Table 6 shows that the SpanProto model experiences improvements across almost all datasets under both 1-shot and 5-shot settings after applying the PA method. In the 1-shot setting, except for a decrease of 0.36% in ACE05, the F_1 scores of the SpanProto model improved by 2.42%, 1.28%, 5.40%, 7.18%, and 5.63% for ACE04, GENIA, ACE05_Chinese, GermEval, and NEREL datasets, respectively, after applying the PA method. In the 5-shot setting, the F_1 scores of the SpanProto model improved by 4.57%, 3.75%, 1.28%, 3.15%, 5.96%, and 3.96% for ACE04, ACE05, GENIA, ACE05_Chinese, GermEval, and NEREL datasets, respectively, after applying the PA mechanism.

The effectiveness of the PA method is not as pronounced for the ESD model compared to SpanProto. After applying the PA mechanism, ESD performs poorly on the ACE04 and ACE05 datasets but shows improvements on other datasets. Specifically, under the 1-shot setting, ESD with PA exhibits improvements on the GENIA, ACE05_Chinese, GermEval, and NEREL datasets by 9.58%, 1.98%, 1.87%, and 7.28%, respectively. Similarly, under the 5-shot setting, ESD with PA demonstrates improvements on the GENIA, ACE05_Chinese, GermEval, and NEREL datasets by 13.70%, 1.92%, 14.65%, and 11.78%, respectively.

F Efficiency Analysis

	finetuning	test
sentence num	50	18496
PACL w/o PA	258.81	342.78
PACL	263.23 (+ 4.42)	343.25 (+ 0.47)
SpanProto	90.32	370.70
w PA	91.68 (+ 1.36)	371.67 (+ 0.97)
ESD	268.21	528.89
w PA	270.97 (+ 2.76)	578.60 (+ 49.71)

Table 8: Our PACL and the two baseline models (ESD and SpanProto) took a certain amount of time for finetuning and testing on the GENIA dataset after applying the PA method (s).

Table 8 displays the time taken by PACL and baseline models. We take the GENIA dataset as an example. Under the 5-shot setting, we extracted 50 sentences along with their labels to form a 32-way 5-shot support set, leaving 18,496 unlabeled

sentences in the unlabeled query set. To measure the time taken for model fine-tuning, we fine-tuned the 50 sentences in the support set for 100 epochs. All models were run on an environment with an Intel Xeon Gold 6348 CPU with a clock speed of 2.60 GHz and an A40 GPU.

We can conclude that SpanProto has the best fine-tuning performance, but our proposed PACL achieves the fastest efficiency during testing. Furthermore, if the models use the PA method, there is not a particularly large loss in time performance. During the fine-tuning stage, in which PACL, SpanProto, and ESD models were trained for 100 epochs on the few-shot support set, using the PA method only resulted in an additional time cost of 4.42, 1.36, and 2.76 seconds, respectively. During testing on the unlabeled query set, using the PA method for testing 18k sentences incurred an additional time cost of only 0.47, 0.97, and 49.71 seconds, respectively.

G Case Study

Figure 5: Sentences from the ACE04, GermEval, and ACE05_Chinese datasets, we present true entities and predicted entities. Entity types are indicated by colored parentheses and background, with the entity categories displayed in corresponding colors above/below the color block.

Figure 5 illustrates test instances of our PACL model on ACE04, GermEval, and ACE05_Chinese datasets. For the example from the ACE04 dataset, PACL missed the entities "Energy Secretary" and "Energy Secretary Bill Richardson." For the example from the GermEval dataset, PACL correctly identified all entities. In the case of the sentence in the ACE05_Chinese dataset, PACL correctly identified the majority of entities but misclassified one entity type (PER classified as GPE).