A Self-Denoising Model for Robust Few-Shot Relation Extraction

Liang Zhang[1,3], Yang Zhang[1], Ziyao Lu[2], Fandong Meng[2], Jie Zhou[2], Jinsong Su[1,3] †
[1]School of Informatics, Xiamen University, China
[2]Pattern Recognition Center, WeChat AI, Tencent Inc, China
[3]Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan (Xiamen University), Ministry of Culture and Tourism, China
lzhang@stu.xmu.edu.cn, jssu@xmu.edu.cn

Abstract

The few-shot relation extraction (FSRE) aims at enhancing the model's generalization to new relations with a few labeled instances. Existing studies usually adopt prototype networks (ProtoNets) for FSRE and assume that the support set, adapting the model to new relations, only contains accurately labeled support instances. However, this assumption is often unrealistic, as even carefully-annotated datasets commonly contain mislabeled instances. In this paper, we first conduct a preliminary study that reveals the high sensitivity of ProtoNets to noisy labels in the support set. Moreover, we find that fully leveraging mislabeled support instances is crucial for enhancing the FSRE model robustness. Thus, we propose a self-denoising model for FSRE, designed to improve model robustness by automatically correcting mislabeled support instances. It comprises two core components: 1) a label correction module (LCM), used to correct noisy labels of support instances based on the distances between them in the embedding space, and 2) a relation classification module (RCM), aimed to achieve more accurate predictions for new relations based on the corrected labels produced by LCM. Moreover, we propose a *feedback-based training strategy*, which focuses on training LCM and RCM to synergistically handle noisy labels in support set. Experimental results on two public datasets confirm the robustness of our model. Particularly, even in scenarios without noisy labels, our model significantly outperforms all baselines.

1 Introduction

Relation extraction (RE), as a crucial information extraction task, has been widely used to provide useful information for various downstream natural language processing (NLP) tasks (Han et al., 2020; Guo et al., 2021). It aims to identify the semantic relation between entities in a given text. In this regard, dominant studies usually employ supervised learning methods, where a large amount of labeled data is used to train various carefully-designed neural networks. Despite their impressive results, these methods face challenges in adapting to new relations (not seen during training). Thus, many researchers have shifted their attention to few-shot relation extraction (FSRE), which focuses on enhancing the generalizability of RE models to new relations with a handful of labeled instances.

Relation 1:
1. The highest elevation in the [**Innerste Uplands**] is the 359m high Griesberg in the [**Hildesheim Forest**].
2. The [**Lamadaya**] waterfalls is one of the best scenery in [**Cal Madow**].
3. Contrary to some literature data, the [**Vršač Mountains**] are notpart of the Carpathians but are a [**Pannonian island mountain**] according to their geotectonic position and geological structure.

Relation 2:
1. All the major dams controlling the [**Nile**] (the [**Old Aswan Dam**] and Esna, Nag Hammadi, Assiut, Edfeina and Zifta barrages) were constructed during this period.
2. The hill on the opposite side of the gorge to the west, the [**Wittekindsberg**], which is the eastern guardian of the [**Wiehen Hills**], defines the western side of this gorge.
3. The westernmost part of the Bavarian Prealps is formed by the [**Ester Mountains**] and its highest peak, the [**Krottenkopf**], which is also the highest summit in the Prealps.

Figure 1: A 2-way-3-shot support set with noisy labels from FewRel 1.0. [**bold**] indicates entities. Can you identify these two relations? Which instances are mislabeled? See **Appendix A** for answers and more cases.

Typically, FSRE studies are conducted in an N-way-K-shot setting, where models are trained and tested on a series of few-shot tasks. Here, each few-shot task contains N new relations, each with K support instances (as support set $\mathcal{S}$) and Q query instances (as query set $\mathcal{Q}$). For each few-shot task, models first adapt to its new relations using the support instances, and then predict the relations of query instances. This process effectively simulates the application of RE models in real-world scenarios where new relations constantly emerge,

This work is done when Liang Zhang was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China.
†Corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26782–26797

each with a few labeled instances. However, prior studies (Han et al., 2018; Gao et al., 2019c; Li et al., 2022; Borchert et al., 2024) usually assume that the support set in each few-shot task contains only accurately labeled support instances. Unfortunately, noisy labels are often inevitable in practical applications (Tsipras et al., 2020; Northcutt et al., 2021) due to label similarity, data ambiguity, or human error. In Figure 1, we intuitively illustrate the challenges of accurately annotating every support instance in FSRE. Furthermore, in Appendix A, we find that advanced large language models (LLMs) (e.g., GPT-4) also struggle to discern mislabeled support instances, even with sufficient hints.

More importantly, since each new relation has only a few support instances in the few-shot task, even a single noisy instance may significantly impair the performance of FSRE models. Thus, it is crucial to further consider the robustness of FSRE models to noisy labels in the support set. Indeed, such noisy few-shot learning (NFSL) has been discussed in the CV community (Mazumder et al., 2021; Liang et al., 2022; Zhang et al., 2023a; Que et al., 2024), but it has not yet been explored in FSRE. Notably, unlike traditional noisy label learning (Song et al., 2022) that aims to reduce the impact of noisy labels in the training set on model training, NFSL focuses on leveraging clean training sets to develop a robust (denoising) model that excels in real-world noisy few-shot tasks. In our experiments, we also explore a more realistic scenario involving NFSL with a noisy training set.

In this paper, we conduct a preliminary study to investigate the robustness of prototypical networks (ProtoNets), widely used in existing FSRE studies (Snell et al., 2017; Li et al., 2022; Borchert et al., 2024), to noisy labels in the support set. We observe that ProtoNets are highly sensitive to such noisy labels. Further analysis reveals that fully utilizing mislabeled support instances is crucial for enhancing the FSRE model's robustness.

Based on the above findings, we propose a self-denoising model for FSRE, which can automatically correct noisy labels of support instances to enhance the model's robustness. As shown in Figure 4, our model consists of three components: an encoder, a label correction module (**LCM**), and a relation classification module (**RCM**). For each few-shot task, we first employ the encoder to generate the embedding representation of every instance. Then, LCM is used to correct noisy labels of support instances according to the distances between them in the embedding space. Finally, RCM accurately predicts the relations of query instances based on the corrected labels generated by LCM.

In addition, the limited training data in FSRE poses a significant challenge for developing a robust FSRE model. Hence, we propose a *feedback-based training strategy* that aims at training LCM and RCM to collaboratively handle noisy labels in the support set. Specifically, we first sample a batch of few-shot tasks $\mathcal{B}$ from the training set and inject noisy labels into their support sets to create a batch of noisy few-shot tasks $\mathcal{B}'$. Next, we use LCM to generate corrected labels for support instances in $\mathcal{B}'$ and employ these corrected labels to guide the training of RCM on $\mathcal{B}'$. Intuitively, the more accurate these corrected labels, the better the trained RCM will perform on $\mathcal{B}$. Thus, we utilize the performance of the trained RCM on $\mathcal{B}$ as a reward signal (*feedback*) to train LCM, enabling it to correct mislabeled support instances more accurately. Unlike prior denoising methods (Liang et al., 2022; Que et al., 2024) that directly use instance labels to supervise model denoising training, we apply the above feedback mechanism to guide the denoising training of our model, more fully utilizing label information from the limited training data.

To evaluate the efficacy of our model, we conduct extensive experiments on two public datasets. Experimental results and in-depth analysis demonstrate the robustness of our model. Notably, even in scenarios without noisy labels, our model significantly outperforms all competitive baselines.

2 Preliminary Study

In this section, we conduct a preliminary study to investigate the robustness of Proto-BERT (Snell et al., 2017), a classic BERT-based ProtoNets, to noisy labels in support set. Although Proto-BERT only includes a BERT-based encoder to obtain instance representations, it still achieves competitive performance and has been widely used in recent studies (Zhang et al., 2022b; Borchert et al., 2024).

As shown in Figure 2, we adjust the noise rate in support set and report the accuracy of Proto-BERT on the validation set of FewRel 1.0. We observe that this model is highly sensitive to noisy labels in support set (See **Green** Line). The intuitive reason is that ProtoNets treat each support instance equally when computing relation prototypes, exacerbating the negative impact of mislabeled instances. To verify this, we follow Liang et al. (2022) to remove

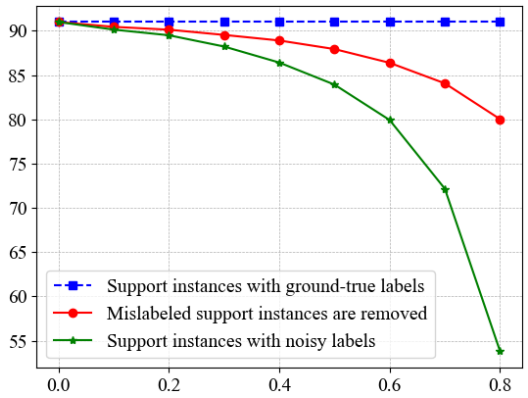


Figure 2: Accuracy of Proto-BERT on the FewRel 1.0 validation set. In the 10-way-10-shot setting, we conduct this group of experiments with different noise rates.

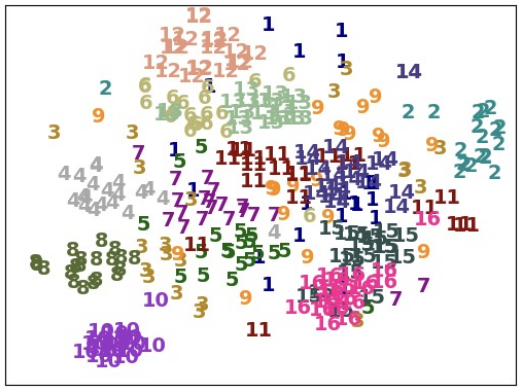


Figure 3: The t-SNE plots of instance representations generated by Proto-BERT on the FewRel 1.0 validation set. The number indicates the relation index.

mislabeled support instances when generating relation prototypes, which improves the model's performance (See **Red** Line). This variant provides a basis for prior studies (Gao et al., 2019a; Liang et al., 2022) that enhance model's robustness by reducing the impact of mislabeled instances on prototype vectors. However, compared to the model using the support instances with ground-truth labels (See **Blue** Line), this variant still exhibits significant performance drop at higher noise rates. This implies that fully utilizing mislabeled support instances is key to further improving the model's robustness.

To better understand the limited robustness of Proto-BERT, we visualize the distribution of instances in its embedding space. From Figure 3, we note that instances of different relations are not distinctly separated. Meanwhile, we use a commonly used clustering metric, the Silhouette Coefficient (Rousseeuw, 1987), to evaluate the discriminability between instances with different relations in the embedding space. In terms of this metric, Proto-BERT achieves a lower score of 0.28, further indicating the insufficient separation between instances of different relations. Intuitively, this insufficient separation worsens the negative impact of mislabeled support instances on relation prototypes.

3 Our Model

In this section, we provide a detailed introduction to our self-denoising model for FSRE. As shown in Figure 4, it contains three components: an encoder, a label correction module (**LCM**), and a relation classification module (**RCM**). We first elaborate these components in Sections 3.1−3.3, and then outline our model training strategy in Section 3.4.

3.1 Encoder

Following prior studies (Liu et al., 2022; Li et al., 2022; Borchert et al., 2024), we utilize BERT (Devlin et al., 2019) as our encoder to generate the embedding representations of each instance in the few-shot task. For each instance x_i, we first insert four special tokens "[E1], [/E1]" and "[E2], [/E2]" into x_i to mark the start and end positions of its head entity and tail entity, respectively. Then, we feed x_i into the encoder to generate its token-level representations. Finally, we concatenate the representations of the [E1] and [E2] tokens at the start positions of the head and tail entities, yielding the representation of this instance: $\mathbf{h}_i=[\mathbf{h}_{[\text{E1}]};\mathbf{h}_{[\text{E2}]}]$.

As per common practices (Liu et al., 2022; Li et al., 2022), we treat the description text of each relation as an additional support instance $x_{i'}$ and apply the encoder to produce its representation $\mathbf{h}_{i'}$. Since $x_{i'}$ does not contain the head and tail entities, we derive its representation by concatenating the representation of the [CLS] token with the average representation of the remaining other tokens: $\mathbf{h}_{i'}=[\mathbf{h}_{[\text{CLS}]};\mathbf{h}_{\text{avg}}]$. Given this support instance, in the N-way-K-shot setting, the support set $\mathcal{S}$ of each few-shot task includes $|\mathcal{S}|=N*(K+1)$ instances.

3.2 Label Correction Module

Back to Figure 4, our LCM comprises two sublayers: a graph neural network (**GNN**) sublayer and an iterative correction (**IC**) sublayer. Particularly, our GNN sublayer is a single-layer graph attention network (GAN) (Veličković et al., 2018), designed to model the correlation among support instances in each few-shot task by graph propagation:

$$\widehat{\mathbf{h}}_i = \sigma\Big(\textstyle\sum_{j=1}^{|\mathcal{S}|} \alpha_{i,j}\mathbf{W}\mathbf{h}_j\Big), \qquad (1)$$

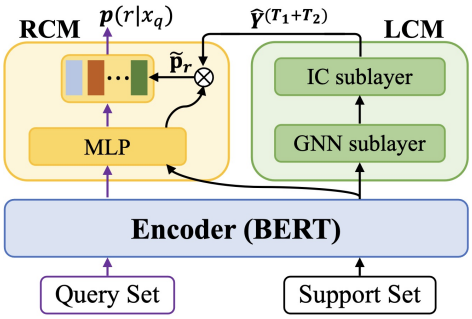


Figure 4: The overall architecture of our model.

where $\alpha_{i,j}$ denotes the attention score between the i-th and j-th support instances, $\mathbf{W}$ is a learnable weight matrix, and $\sigma(\cdot)$ refers to the activation function (i.e., ReLU). With the help of our training strategy, this sublayer can map the encoder-generated representations of support instances into a more discriminative embedding space, where instances with the same relation are clustered and those with different relations are separated. This lays the foundation for correcting noisy labels of support instances and is validated in our experiments.

On top of the GNN sublayer, the IC sublayer aims to iteratively correct noisy labels of support instances based on the distances between them in the embedding space. Here, we sequentially adopt two iterative strategies to gradually refine the labels of support instances:

The first strategy. It focuses on leveraging the distances among support instances with the same relation to iteratively correct noisy labels in the support set. Specifically, at the t_1-th iteration, we first compute an importance weight $w_i^{(t_1)}$ for each support instance based on its distance to the nearest support instance sharing the same relation with it:

$$w_i^{(t_1)} = -\min_{j\in\mathcal{N}_r, j\neq i}\{\|\widehat{\mathbf{h}}_i - \widehat{\mathbf{h}}_j\|_2\}, \quad (2)$$

where $r{=}\widehat{y}_i^{(t_1-1)}$ denotes the relation generated in the previous iteration for the i-th support instance, $\mathcal{N}_r$ symbolizes the index set of all support instances with the relation r, and $\|\cdot\|_2$ represents the L2 distance function. Intuitively, mislabeled support instances often have smaller importance weights than correctly labeled ones, as they are generally more distant from other instances with the same relation. Next, we calculate a prototype vector $\mathbf{p}_r^{(t_1)}$ for each relation r using the weighted average representation of its support instances:

$$\mathbf{p}_r^{(t_1)} = \sum_{j\in\mathcal{N}_r} \frac{\exp(w_j^{(t_1)})}{\sum_{j'\in\mathcal{N}_r}\exp(w_{j'}^{(t_1)})}\widehat{\mathbf{h}}_j. \quad (3)$$

In this way, we can lessen the effect of mislabeled support instances on the relation prototypes. Finally, we use these prototype vectors to correct noisy labels of support instances as follows:

$$\widehat{y}_i^{(t_1)} = \arg\max_r(-\|\widehat{\mathbf{h}}_i - \mathbf{p}_r^{(t_1)}\|_2), \quad (4)$$

where $\widehat{y}_i^{(t_1)}$ is the updated label of the i-th support instance. Through repeating the above process T_1 times, we can obtain corrected labels $\widehat{\mathbf{Y}}^{(T_1)}$ for all support instances, where each row in $\widehat{\mathbf{Y}}^{(T_1)}$ denotes the one-hot label of a support instance.

The second strategy. It aims to further refine the labels of support instances via a label propagation manner, which iteratively updates the label of each support instance using the label information of its neighboring instances in the embedding space. Concretely, we first construct an adjacency matrix $\mathbf{A}{=}[a_{i,j}]{\in}\mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ using the distances $a_{i,j}{=}\widehat{\mathbf{h}}_i^{\mathsf{T}}\widehat{\mathbf{h}}_j$ between support instances. Meanwhile, we set the diagonal elements of $\mathbf{A}$ to $-\infty$ and normalize its row vectors via a softmax function. Lastly, we perform label propagation by iterating the following process:

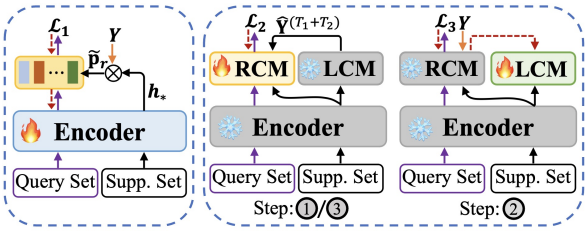
$$\widehat{\mathbf{Y}}^{(T_1+t_2)} = \mathbf{A}\widehat{\mathbf{Y}}^{(T_1+t_2-1)} + \widehat{\mathbf{Y}}^{(T_1+t_2-1)}, \quad (5)$$

where t_2 indicates the current iteration step. Unlike traditional label propagation methods, we mask the diagonal elements of $\mathbf{A}$ and retain the label information from the previous iteration by a residual connection. After T_2 iterations, we obtain refined corrected labels $\widehat{\mathbf{Y}}^{(T_1+T_2)}$ for all support instances.

The primary motivation behind combining the above two iterative strategies is that the first strategy is more effective at high noise rates while the second strategy performs better at low noise rates (as confirmed in ablation studies). Thus, combining them enables our model to correct noisy labels of support instances across various noise levels.

3.3 Relation Classification Module

Our RCM consists of a simple multi-layer perceptron (MLP), which aims to predict the relations of query instances based on support instances with the corrected labels $\widehat{\mathbf{Y}}^{(T_1+T_2)}$. As shown in the top left of Figure 4, we first feed the encoder-generated instance representation $\mathbf{h}_i$ into the MLP, yielding its updated representation $\widetilde{\mathbf{h}}_i{=}\mathrm{MLP}(\mathbf{h}_i)$. Next, we normalize the column vectors of $\widehat{\mathbf{Y}}^{(T_1+T_2)}$ to obtain the relevance score $s_{i,r}$ between each support



(1) Pre-training stage **(2)** Self-denoising stage

Figure 5: Illustration of our model training. Red dashed vectors denote gradients.

instance and each relation. With these scores, we derive a prototype vector $\widetilde{\mathbf{p}}_r$ for each relation r:

$$\widetilde{\mathbf{p}}_r = \sum_{i=1}^{|\mathcal{S}|} s_{i,r}\widetilde{\mathbf{h}}_i = \sum_{i=1}^{|\mathcal{S}|} \frac{\widehat{\mathbf{Y}}_{i,r}^{(T_1+T_2)}}{\sum_{j=1}^{|\mathcal{S}|}\widehat{\mathbf{Y}}_{j,r}^{(T_1+T_2)}}\widetilde{\mathbf{h}}_i. \quad (6)$$

These prototype vector, benefiting from $\widehat{\mathbf{Y}}^{(T_1+T_2)}$, is more robust to mislabeled support instances.

Following this, we can compute the relation distribution $\widetilde{p}(r|x_q)$ of each query instance:

$$\widetilde{p}(r|x_q) = \frac{\exp(-\|\widetilde{\mathbf{h}}_q - \widetilde{\mathbf{p}}_r\|_2)}{\sum_{r'=1}^{N}\exp(-\|\widetilde{\mathbf{h}}_q - \widetilde{\mathbf{p}}_{r'}\|_2)}. \quad (7)$$

Finally, we predict the relation $\widetilde{y}_q$ for each query instance x_q as follows: $\widetilde{y}_q = \arg\max_r (\widetilde{p}(r|x_q))$.

3.4 Model Training

Our model undergoes two training stages: a pre-training stage and a self-denoising stage. At the first stage, we define a loss function $\mathcal{L}_1$ that is only related to our encoder and use it to pre-train the encoder on few-shot tasks sampled from the training set. As shown in Figure 5 (1), we first replace $\widehat{\mathbf{Y}}^{(T_1+T_2)}$ and $\{\widetilde{\mathbf{h}}_*\}$ in Eqs. (6-7) with the ground-truth labels $\mathbf{Y}$ of support instances and the encoder-generated instance representations $\{\mathbf{h}_*\}$, yielding a new relation distribution $p(r|x_q)$ for each query instance. Then, we formalize $\mathcal{L}_1$ as follows:

$$\mathcal{L}_1 = \sum_{q=1}^{|\mathcal{Q}|} -\log(p(y_q|x_q)), \quad (8)$$

where $\mathcal{Q}$ denotes the query set, y_q refers to the ground-truth label of the q-th query instance x_q.

At the self-denoising stage, we freeze the trained encoder and propose a *feedback-based training strategy* to train our RCM and LCM, allowing them to collaboratively handle noisy labels in the support set. Concretely, we first sample a batch of few-shot tasks $\mathcal{B}$ from the training set and inject noisy labels into their support sets, obtaining a batch of noisy few-shot tasks $\mathcal{B}'$. Then, we alternately trains RCM and LCM on $\mathcal{B}'$ and $\mathcal{B}$ by three training steps:

In the first step, we focus on applying LCM to guide the training of RCM on $\mathcal{B}'$. As depicted in Figure 5 (2), we first employ LCM to generate the corrected labels $\widehat{\mathbf{Y}}^{(T_1+T_2)}$ for support instances in $\mathcal{B}'$. Next, we feed $\widehat{\mathbf{Y}}^{(T_1+T_2)}$ into RCM to predict the relation distribution $\widetilde{p}(r|x_q)$ for each query instance in $\mathcal{B}'$ based on Eqs. (6-7), further obtaining a new loss $\mathcal{L}_2 = \sum_{q=1}^{|\mathcal{Q}|} -\log(\widetilde{p}(y_q|x_q))$. Lastly, we use the adapted Adam optimizer (Pham et al., 2021; Ding et al., 2022) and the loss $\mathcal{L}_2$ to optimize our RCM, resulting in a temporary RCM. Since $\mathcal{L}_2$ is derived from $\widehat{\mathbf{Y}}^{(T_1+T_2)}$ generated by LCM, this optimizer can ensure the dependency between the temporary RCM parameters and the LCM ones, enabling the loss computed by the temporary RCM to directly calculate the gradient of the LCM parameters. Intuitively, the more accurate the corrected labels $\widehat{\mathbf{Y}}^{(T_1+T_2)}$ produced by LCM, the better the temporary RCM will perform on $\mathcal{B}$.

In the second step, inspired by the above intuition, we utilize the performance of the temporary RCM on $\mathcal{B}$ as a reward signal (*feedback*) to train our LCM. As shown in the right part of Figure 5 (2), we input the ground-truth labels $\mathbf{Y}$ of support instances in $\mathcal{B}$ into the temporary RCM to re-predict a new relation distribution $\hat{p}(r|x_q)$ for each query instance in $\mathcal{B}$ (or $\mathcal{B}'$), thus deriving our third loss $\mathcal{L}_3 = \sum_{q=1}^{|\mathcal{Q}|} -\log(\hat{p}(y_q|x_q))$. The loss $\mathcal{L}_3$ reflects the temporary RCM's performance on $\mathcal{B}$ and indirectly measures the accuracy of the corrected labels $\widehat{\mathbf{Y}}^{(T_1+T_2)}$ generated by LCM in the first step. Therefore, we use $\mathcal{L}_3$ to optimize our LCM, enabling it to generate more accurate corrected labels for the mislabeled support instances.

In the third step, we utilize the updated LCM to generate more accurate corrected labels $\widehat{\mathbf{Y}}^{(T_1+T_2)}$ for support instances in $\mathcal{B}'$. Then, the new corrected labels $\widehat{\mathbf{Y}}^{(T_1+T_2)}$ are fed into the original RCM to compute a new $\mathcal{L}_2$ loss, as done in the first step. Finally, we update the RCM parameters with this loss, allowing it to more accurately predict the relations of query instances based on the corrected labels generated by LCM.

Notably, the above training process is efficient, attributed to the limited parameters in our LCM (a single-layer GAN) and RCM (a small MLP). This allows our model to attain training and inference efficiency on par with traditional ProtoNets.

4 Experiment

4.1 Datasets & Evaluation

Our experiments are conducted on two widely used datasets:

- **FewRel 1.0** (Han et al., 2018). It is a large-scale human-annotated dataset constructed from Wikipedia articles. This dataset contains 100 relations, each with 700 instances. We follow official split to use 64 relations for training, 16 for validation, and 20 for testing.
- **FewRel 2.0** (Gao et al., 2019c). This dataset aims to evaluate the domain transferability of FSRE models. It utilizes the same training set as FewRel 1.0, derived from the Wikipedia domain. Meanwhile, it introduces a new test set from the biomedical domain, which contains 25 relations, each with 100 instances.

Notably, in both datasets, the training, validation, and test sets contain mutually exclusive relations.

With the official evaluation script, we evaluate our model on 10,000 few-shot tasks sampled from the validation and test sets. As in previous studies (Han et al., 2018; Gao et al., 2019c), we utilize accuracy as our evaluation metric. Following prior studies (Liang et al., 2022; Zhang et al., 2023a), we create a support set with noisy labels by randomly replacing the ground-truth labels of support instances with other labels from the current few-shot task. Since this type of noisy label often has a greater impact on the FSRE model's performance than others, it can more rigorously evaluate the model's robustness in real-world scenarios.

4.2 Setting

We implement our model based on Huggingface's Transformers (Wolf et al., 2020) and PyTorch (Paszke et al., 2019). To optimize our model, we use AdamW (Loshchilov et al., 2019) as our optimizer, which is equipped with a linear warmup (Goyal et al., 2017) for the first 10% training steps. The learning rate of our encoder is set to 1e-5, while that of other modules is set to 5e-5. In the N-way-K-shot setting, we set the value of N to 5 and 10, and K to 5. In the 10-way-5-shot setting, it takes 5.5 hours to train our model on FewRel 1.0 with a single 24GB NVIDIA RTX 3090 GPU.

4.3 Baselines

We compare our model with the following baselines: 1) **Proto-BERT** (Snell et al., 2017) is a ProtoNet only equipped with a BERT-based encoder, serving as our base model. 2) **SimpleFSRE** (Liu et al., 2022) utilizes the description text of each relation as an additional support instance to improve its prototype vector. 3) **LPD** (Zhang et al., 2022b) adopts a *label prompt dropout* method to prevent the model from overfitting to the relation description text. 4) **GM_GEN** (Li et al., 2022) uses a GNN to model the correlation among support instances with the same relation, yielding better relation prototypes. 5) **MultiRep** (Borchert et al., 2024) combines multiple sentence representations to obtain relation prototypes by contrastive learning. 6) **Proto-BERT-Remove** is a variant of Proto-BERT, which enhances the model's robustness by removing mislabeled support instances when computing relation prototypes. 7) **HATT-BERT** (Gao et al., 2019a) employs an attention mechanism to mitigate the negative impact of mislabeled support instances on relation prototypes. It aims to reduce the impact of unknown noise in the training set on model training, and is therefore used to pre-train our encoder when unknown noises exist in the training set (See **Appendix D.3**). 8) **RNNP** (Mazumder et al., 2021) refines prototype vectors by producing hybrid features from support instances. 9) **TraNFS** (Liang et al., 2022) obtains robust prototype vectors by utilizing the Transformer's encoder layer to aggregate support instances. Moreover, we compare our model with several several LLM-based baselines, including **CoT-ER** (Ma et al., 2023) and **ICRE** (Li et al., 2024). Here, we reimplement CoT-ER using ChatGPT (OpenAI, 2022) alongside SBERT (Reimers et al., 2019), and reproduce ICRE on Llama-2 (Touvron et al., 2023).

4.4 Main Results

Results on FewRel 1.0. Experimental results on the validation and test sets of FewRel 1.0 are shown in Table 1. We observe that our model consistently outperforms all baselines across all settings. Moreover, we draw several conclusions:

The first part of Table 1 shows the performance of recent competitive baselines in FSRE, which do not consider the issue of noisy labels. Despite achieving improved results at a 0% noise rate, these baselines are highly sensitive to noisy labels in the support set, limiting their practicality. This also indicates that enhancing the FSRE model's robustness to noisy labels in the support set is crucial.

In the second part of Table 1, we report the performance of several classic denoising models in

N-way-K-shot	5-way-5-shot					10-way-5-shot				
	val				test	val				test
Noise Rate	0%	20%*	40%*	60%*	0%	0%	20%*	40%*	60%*	0%
Proto-BERT (Snell et al., 2017)	91.32	87.27	82.21	70.72	89.60	83.68	80.35	72.55	57.27	82.89
LPD (Zhang et al., 2022b)	90.65	89.22	84.31	74.19	95.07	82.15	81.08	73.89	61.30	91.08
SimpleFSRE (Liu et al., 2022)	94.05	89.80	84.77	74.43	96.37	89.68	83.11	74.87	61.51	93.47
GM_GEN (Li et al., 2022)	95.62	89.60	84.58	74.60	96.96	91.27	82.67	74.60	61.94	94.30
MultiRep (Borchert et al., 2024)	93.79	89.57	84.66	74.51	96.29	88.80	82.90	74.88	61.75	91.98
Proto-BERT-Remove (Snell et al., 2017)*	91.32	88.89	87.71	80.88	89.60	82.89	81.14	79.85	72.16	83.68
HATT-BERT (Gao et al., 2019a)*	94.26	92.46	90.28	80.19	96.42	89.74	87.24	84.17	71.95	93.51
RNNP (Mazumder et al., 2021)*	94.26	92.46	90.28	80.19	96.42	89.74	87.24	84.17	71.95	93.51
TraNFS (Liang et al., 2022)*	93.87	90.92	89.07	79.93	96.11	89.17	86.70	83.98	71.82	93.03
CoT-ER (+ChatGPT) (Ma et al., 2023)*	94.07	89.38	83.26	73.11	96.59	89.50	82.54	73.64	58.82	93.25
MICRE (+Llama-2) (Li et al., 2024)*	93.90	89.22	84.57	73.81	96.46	88.87	82.73	74.25	61.01	92.24
Ours	**96.68**	**96.33**	**95.41**	**91.22**	**97.33**	**92.17**	**91.88**	**90.23**	**86.22**	**94.96**

Table 1: Accuracy (%) on the FewRel 1.0 validation and test sets. Results with * are obtained by our reproduction.

NFSL, which are designed to reduce the impact of mislabeled support instances on relation prototypes. In comparison to these models, our model consistently exhibits greater robustness across various noise rates. It suggests that our model can further enhance its robustness by correcting mislabeled support instances (thus fully utilizing them).

As shown in the third part of Table 1, these LLM-based FSRE models also exhibit high sensitivity to noisy labels in the support set. These findings underscore the significance of improving the FSRE model's robustness to noisy labels in the support set, even when leveraging LLMs for FSRE.

Particularly, our model still significantly outperforms all competitive baselines at a 0% noise rate. Notably, in this case, our model is tested on clean few-shot tasks, while our feedback-based training strategy utilizes noisy few-shot tasks with the lowest noise rate (20%) to train our LCM and RCM. There are two underlying reasons for this: 1) the feedback mechanism in our training strategy can more effectively utilize the label information in the limited training set to train our model, leading to improved overall model performance. 2) the noisy few-shot tasks employed in our training strategy function as a data augmentation, mitigating the scarcity of training data in FSRE.

Results on FewRel 2.0. Similar to its performance on FewRel 1.0, our model consistently outperforms all baselines on FewRel 2.0. These results and detailed analyses are presented in **Appendix C**.

4.5 Ablation Study

We further conduct ablation studies by removing various components of our model to assess their individual contributions. Specifically, we compare our model with the following variants in Table 2:

Model	0%	20%	40%	60%
Ours	**96.68**	**96.33**	**95.41**	**91.22**
1 *w/o.* GNN sublayer	96.03	94.53	93.02	87.78
2 *w/o.* First iterative strategy	96.57	95.08	90.22	78.53
3 *w/o.* Second iterative strategy	95.91	94.86	93.33	88.04
4 *w/o.* Encoder pre-training	96.17	95.28	92.07	85.01
5 *w/o.* FTS	95.55	95.06	93.61	88.68
6 *w/o.* RCM & FTS	95.03	94.75	93.11	88.07

Table 2: Ablation results on FewRel 1.0 validation set in the 5-way-5-shot setting. FTS is an acronym of Feedback-based Training Strategy.

(1) *w/o. GNN sublayer*. In this variant, we remove the GNN sublayer from our LCM. From **Line 1** of Table 2, we observe that this change leads to a significant performance drop of 3.44 points when the noise rate is set to 60%. This suggests that our GNN sublayer can indeed map support instances into a better embedding space, allowing the IC sublayer to more accurately correct their noisy labels. For further analysis, refer to **Appendix D.3**.

(2) *w/o. First iterative strategy* and *w/o. Second iterative strategy*. In the IC sublayer, we sequentially apply two iterative strategies to gradually correct noisy labels of support instances. In these two variants, we respectively discard these two strategies from our model to validate their efficacy. The results presented in **Line 2** and **Line 3** indicate that the first strategy is more effective under high noise conditions, while the second strategy performs better at low noise rates. Moreover, our model achieves further performance gains by

combining these two strategies, which effectively demonstrates their compatibility.

(3) *w/o. Encoder pre-training*. This variant directly employs our feedback-based training strategy to jointly train all components of our model. In this context, the encoder is regarded as a part of our LCM and is updated along with it. As illustrated in **Line 4**, this variant exhibits a significant performance decline, particularly at high noise rates. The potential reason is that noisy labels in the support set negatively affect the expressive ability of our encoder during training. Furthermore, compared to our model, this variant significantly increases the model training time (5.5 hours vs. 13.5 hours). These results effectively confirm that encoder pre-training is crucial for improving the efficiency and stability of our model training.

(4) *w/o. FTS*. Here, we remove the feedback-based training strategy from our model and only use $\mathcal{L}_2$ in Section 3.4 to jointly train our LCM and RCM, which leads to a significant performance drop across all settings (See **Line 5**). This reveals that our feedback-based training strategy can more effectively train our LCM and RCM to collaboratively handle noisy labels of support instances.

(5) *w/o. RCM & FTS*. Following prior denoising methods (Gao et al., 2019a; Liang et al., 2022), this variant directly utilizes the ground-truth labels of support instances to supervise the training of our LCM on noisy few-shot tasks, while excluding the feedback-based training strategy and RCM from our model. According to **Line 6**, this alteration consistently degrades our model's performance across all noise levels, particularly at a 0% noise rate. These results further validate that our training strategy can indeed more efficiently exploit the label information in the limited training set to train our model, thus improving the robustness and overall performance of our model.

Notably, the hyper-parameter settings of our model are discussed in **Appendix B**. Moreover, in **Appendix D**, we further analyze our model's correction capability and its robustness to unknown noise in the training set, while also intuitively illustrating the effectiveness of our GNN sublayers.

5 Related Work

Relation extraction (RE) is a fundamental task in information extraction, which aims to identify the semantic relation between two entities in a given text (Zeng et al., 2020; Han et al., 2020; Zhang et al., 2022a; Chen et al., 2022; Zhang et al., 2023b,c, 2024; Yue et al., 2024). However, conventional methods heavily rely on a large number of labeled data and struggle with handling unseen relations. Therefore, recent studies turn to few-shot relation extraction (FSRE) that aims to train a model to classify instances into novel relations with only a few labeling instances.

Most existing studies (Gao et al., 2019b; Yang et al., 2020; Han et al., 2021; Zhang et al., 2022b, 2023d) apply prototypical networks for FSRE, which aims to learn a suitable prototype vector for each relation from a handful of labeling instances. For example, Ye et al. (2019) present a multi-level matching and aggregation network to interactively encode query and support instances by considering their matching information at both local and instance levels. Qu et al. (2020) proposes a Bayesian meta-learning approach to learn the correlations among different relations. Zeng et al. (2020) distinguishes between hard and easy few-shot tasks and proposes a hybrid contrastive relation-prototype method to improve model performance on hard few-shot tasks. Meanwhile, some studies (Yang et al., 2020; Wang et al., 2020; Han et al., 2021; Borchert et al., 2024) aim to obtain more expressive relation prototypes by utilizing additional information such as entity and relation description texts. Recently, Li et al. (2022) achieve new state-of-the-art performance by leveraging a GNN to learn better relational prototypes.

Moreover, some studies (Dong et al., 2021) focus on further training pre-trained language models (PLMs) using noisy RE datasets to improve the model's generalization to new relations. Soares et al. (2019) propose a matching the blanks method to further train PLMs on their collected dataset. Peng et al. (2020) introduce an entity-masked contrastive pre-training framework for FSRE. Zhang et al. (2022b) establish a more rigorous pre-training setting by filtering relations contained in FewRE 1.0 from the pre-trained corpus.

Notably, these methods usually assume that the support set, adapting the model to new relations, only includes accurately labeled instances. However, this assumption is usually unrealistic because even carefully-annotated dataset often contains mislabeled instances. Inspired by noisy few-shot learning in the computer vision community (Mazumder et al., 2021; Liang et al., 2022; Zhang et al., 2023a; Que et al., 2024), we introduce a more realistic testing setting for FSRE, where the support set in

test few-shot tasks contains noisy labels. In this way, we can further evaluate the practicality and robustness of FSRE models. Meanwhile, we propose a self-denoising model for FSRE, which can automatically correct noisy labels of support instances to enhance the robustness of the model.

6 Conclusion and Future Work

In this paper, we propose a self-denoising model for FSRE, which focuses on automatically correcting noisy labels of support instances to enhance the model robustness. Our model comprises two core component: a label correction module (LCM) and a relation classification module (RCM). For each few-shot task, we first use LCM to iteratively correct noisy labels of support instances. Then, we feed these corrected labels into RCM to generate more robust relation prototypes, thus more accurately predicting the relation of each query instances. Moreover, we propose a feedback-based training strategy, designed to train our LCM and RCM to synergistically handle noisy labels of support instances. Experimental results and in-depth analysis on two public datasets demonstrate the effectiveness and robustness of our model.

In the future, we plan to apply our model to other tasks involving noisy labels, such as text and image classification tasks, so as to verify its generality.

Acknowledgments

The project was supported by National Natural Science Foundation of China (No. 62276219), Natural Science Foundation of Fujian Province of China (No. 2024J011001), and the Public Technology Service Platform Project of Xiamen (No.3502Z20231043). We also thank the reviewers for their insightful comments.

Limitations

The limitations of our method mainly include following two aspects: 1) Our method is only examined on the FSRE task, while whether it is able to generalize to other tasks, such as image classification, is not yet explored in this paper. 2) We did not consider non-of-the-above scenarios where a query instance may not belong to any class in the few-shot task.

References

Philipp Borchert, Jochen De Weerdt, Marie-Francine Moens, Philipp Borchert, Jochen De Weerdt, and Marie-Francine Moens. 2024. Efficient information extraction in few-shot relation classification through contrastive representation learning. In *NAACL*.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *WWW*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Kaize Ding, Jianling Wang, James Caverlee, and Huan Liu. 2022. Meta propagation networks for graph few-shot semi-supervised learning. In *AAAI*.

Manqing Dong, Chunguang Pan, Zhipeng Luo, Manqing Dong, Chunguang Pan, and Zhipeng Luo. 2021. Mapre: An effective semantic mapping approach for low-resource relation extraction. In *EMNLP*.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019a. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *AAAI*.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019b. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *AAAI*.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019c. Fewrel 2.0: Towards more challenging few-shot relation classification. In *EMNLP*.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. In *arXiv:1706.02677*.

Zhijiang Guo, Guoshun Nan, Wei Lu, and Shay B Cohen. 2021. Learning latent forests for medical relation extraction. In *IJCAI*.

Jiale Han, Bo Cheng, and Wei Lu. 2021. Exploring task difficulty for few-shot relation extraction. In *EMNLP*.

Jiale Han, Bo Cheng, Xizhou Wang, Bo Cheng, and Xizhou Wang. 2020. Two-phase hypergraph based reasoning with dynamic relations for multi-hop kbqa. In *IJCAI*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *EMNLP*.

Guozheng Li, Peng Wang, Jiajun Liu, Yikai Guo, Ke Ji, Ziyu Shang, and Zijie Xu. 2024. Meta in-context learning makes large language models better zero and few-shot relation extractors. In *IJCAI*.

Wanli Li, Tieyun Qian, Wanli Li, and Tieyun Qian. 2022. Graph-based model generation for few-shot relation extraction. In *EMNLP*.

Kevin J Liang, Samrudhdhi B Rangrej, Vladan Petrovic, and Tal Hassner. 2022. Few-shot learning with noisy labels. In *CVPR*.

Yang Liu, Jinpeng Hu, Xiang Wan, and Tsung-Hui Chang. 2022. A simple yet effective relation information guided approach for few-shot relation extraction. In *Findings of ACL*.

Ilya Loshchilov, Frank Hutter, Ilya Loshchilov, and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR 2019*.

Xilai Ma, Jing Li, Min Zhang, Xilai Ma, Jing Li, and Min Zhang. 2023. Chain of thought with explicit evidence reasoning for few-shot relation extraction. In *Findings of EMNLP*.

Pratik Mazumder, Pravendra Singh, Vinay P Namboodiri, Pratik Mazumder, Pravendra Singh, and Vinay P Namboodiri. 2021. Rnnp: A robust few-shot learning approach. In *WACV*.

Curtis G Northcutt, Anish Athalye, Jonas Mueller, Curtis G Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. *NeurIPS Datasets and Benchmarks*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. In *OpenAI Blog*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? an empirical study on neural relation extraction. In *ACL*.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. 2021. Meta pseudo labels. In *CVPR*.

Meng Qu, Tianyu Gao, Louis-Pascal Xhonneux, and Jian Tang. 2020. Few-shot relation extraction via bayesian meta-learning on relation graphs. In *ICML*.

Xiaofan Que, Qi Yu, Xiaofan Que, and Qi Yu. 2024. Dual-level curriculum meta-learning for noisy few-shot learning tasks. In *AAAI*.

Nils Reimers, Iryna Gurevych, Nils Reimers, and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *In J. Comput. Appl. Math.*

Jake Snell, Kevin Swersky, Richard Zemel, Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NIPS*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *ACL*.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE Trans. Neural Networks Learn. Syst.*

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. In *arXiv:2307.09288*.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. 2020. From imagenet to image classification: Contextualizing progress on benchmarks. In *ICML*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. *In ICLR*.

Yuxia Wang, Karin Verspoor, and Timothy Baldwin. 2020. Learning from unlabelled data for clinical semantic textual similarity. In *ClinicalNLP*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP*.

Kaijia Yang, Nantao Zheng, Xinyu Dai, Liang He, Shujian Huang, and Jiajun Chen. 2020. Enhance prototypical network with text descriptions for few-shot relation classification. In *CIKM*.

Zhi-Xiu Ye, Zhen-Hua Ling, Zhi-Xiu Ye, and Zhen-Hua Ling. 2019. Multi-level matching and aggregation network for few-shot relation classification. In *ACL*.

Hao Yue, Shaopeng Lai, Chengyi Yang, Liang Zhang, Junfeng Yao, and Jinsong Su. 2024. Towards better graph-based cross-document relation extraction via non-bridge entity enhancement and prediction debiasing. In *Findings of ACL*.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *EMNLP*.

Ji Zhang, Lianli Gao, Xu Luo, Hengtao Shen, and Jingkuan Song. 2023a. Deta: Denoised task adaptation for few-shot learning. In *ICCV*.

Liang Zhang, Zijun Min, Jinsong Su, Pei Yu, Ante Wang, and Yidong Chen. 2023b. Exploring effective inter-encoder semantic interaction for document-level relation extraction. In *IJCAI*.

Liang Zhang, Jinsong Su, Yidong Chen, Zhongjian Miao, Zijun Min, Qingguo Hu, and Xiaodong Shi. 2022a. Towards better document-level relation extraction via iterative inference. In *EMNLP*.

Liang Zhang, Jinsong Su, Zijun Min, Zhongjian Miao, Qingguo Hu, Biao Fu, Xiaodong Shi, and Yidong Chen. 2023c. Exploring self-distillation based relational reasoning training for document-level relation extraction. In *AAAI*.

Liang Zhang, Zhen Yang, Biao Fu, Ziyao Lu, Liangying Shao, Shiyu Liu, Fandong Meng, Jie Zhou, Xiaoli Wang, and Jinsong Su. 2024. Multi-level cross-modal alignment for speech relation extraction. In *EMNLP*.

Liang Zhang, Chulun Zhou, Fandong Meng, Jinsong Su, Yidong Chen, and Jie Zhou. 2023d. Hypernetwork-based decoupling to improve model generalization for few-shot relation extraction. In *EMNLP*.

Peiyuan Zhang, Wei Lu, Peiyuan Zhang, and Wei Lu. 2022b. Better few-shot relation extraction with label prompt dropout. In *EMNLP*.

$T_2 \backslash T_1$	1	2	3	4	5	6
1	88.86	89.15	89.83	90.16	90.01	90.06
2	89.26	89.82	90.07	90.39	90.17	90.21
3	89.71	89.64	90.23	90.51	90.66	90.47
4	89.91	90.75	90.46	90.53	90.21	90.61
5	90.28	90.43	90.61	90.67	90.32	90.55
6	90.19	90.50	90.79	**91.22**	90.72	90.51
7	90.25	90.24	90.59	91.10	91.02	90.84
8	90.37	90.35	90.63	90.89	90.53	90.63

Table 3: Accuracy (%) of our model with different values of T_1 and T_2 on the FewRel 1.0 validation set. These experiments are conducted in the 5-way-5-shot setting at a 60% noise rate.

A Further Analysis of Noisy Support Sets

In this section, we aim to further elaborate on the fact that noise labels are inevitable in FSRE. First, we apply the popular large language models (i.e., Chat-GPT and GPT-4) to identify the relations and mislabeled instances in the case illustrated in Figure 1. As shown in Figure 7, even with sufficient hints, both Chat-GPT and GPT-4 fail to accurately predict these two relations and mislabeled support instances. This intuitively illustrates the challenge of identifying mislabeled support instances from the support set, and highlights the inevitability of noisy labels in the support set.

Meanwhile, Figure 8 presents two more representative cases of support sets with noisy labels, both of which are real cases from FewRel 1.0 and FewRel 2.0, respectively. Due to space limitations, we provide only a simplified version of **Case 1** in Figure 1. It is worth noting that the second case comes from the biomedical field. From these two cases, we drew several conclusions: (1) The similarity among relations is one of the factors that leads to noisy labels in FSRE (see **Case 1** (2) In some specialized domains, such as biomedicine and finance, obtaining accurate relation label for every support instance is greater challenging due to the substantial expertise required (See **Case 2**). (3) By observing the instances in Figure 8, it is intuitively evident that acquiring accurate labeled data for RE is more challenging than the image classification tasks in the CV community. However, noisy few-shot learning has been widely discussed in the CV community, but it has not yet to be explored in the context of FSRE. Therefore, we hope our work will inspire further research to design more robust and practical FSRE models.

B Hyper-parameter Settings

B.1 Effect of iteration numbers T_1 and T_2

To investigate the impact of hyper-parameters T_1 and T_2 on our model, we compare the performance of models with different values of T_1 and T_2 on the FewRel 1.0 validation set. As illustrated in Table 3, our model achieves the best performance when T_1 and T_2 are set to 4 and 6, respectively. Meanwhile, we also observe that our model exhibits low sensitivity to these two hyper-parameters. Hence, we adopt T_1=4 and T_2=6 for all other experiments.

B.2 Effect of distance function in IC sublayer

In the two iteration strategies of the IC sublayer, we employ L2 distance and dot product to measure the distances between support instances in the embedding space, respectively. To further investigate the impact of different distance functions on our IC sublayer, we present the performance of our model variants with different distance functions. From Table 4, we observe that using different distance functions in these two iteration strategies generally leads to better performance. The underlying reason is that different distance functions can assess the correlation between support instances from diverse perspectives, allowing our IC sublayer to more accurately estimate whether two support instances share the same relation. Notably, our model achieves optimal performance when using L2 distance and dot product in our two iteration strategies, respectively.

C Results on FewRel 2.0

To assess the domain transfer ability of our model, we also conduct experiments on FewRel 2.0. As illustrated in Table 5, our model still consistently outperforms all baselines across all settings, which further demonstrates the superiority of our model. Meanwhile, we note that most models exhibit similar sensitivity to noisy labels in the support set on FewRel 2.0 as they do on FewRel 1.0.

D Further Analysis

D.1 Analysis for correction performance

To further illustrate the correction capability of our model, we report its accuracy in correcting noisy labels of support instances. As shown in Table 6, our model successfully corrects over 90% noisy labels even at a 60% noise rate, which further demonstrates the robustness of our model. Meanwhile,

The distance functions					
First iterative strategy	Second iterative strategy	0%	20%	40%	60%
L2 distance	L2 distance	95.16	94.72	93.84	89.03
L2 distance	Dot product	**96.68**	**96.33**	**95.41**	**91.22**
L2 distance	Cosine similarity	95.83	95.81	94.80	90.53
Dot product	L2 distance	94.47	93.89	93.01	88.39
Dot product	Dot product	94.09	93.21	92.53	87.81
Dot product	Cosine similarity	94.11	93.39	92.70	87.97
Cosine similarity	L2 distance	94.26	93.41	92.83	88.13
Cosine similarity	Dot product	94.02	93.09	92.40	87.55
Cosine similarity	Cosine similarity	93.78	92.79	92.02	87.26

Table 4: Accuracy (%) on the FewRel 1.0 validation set in the 5-way-5-shot setting.

N-way-K-shot	5-way-5-shot				10-way-5-shot			
Noise Rate	0%	20%*	40%*	60%*	0%	20%*	40%*	60%*
Proto-BERT (Snell et al., 2017)	51.50	50.37	47.16	34.63	36.93	36.03	30.95	17.79
LPD (Zhang et al., 2022b)	86.90	82.86	73.97	66.73	78.43	74.93	66.56	56.50
GM_GEN (Li et al., 2022)	91.28	88.61	82.48	67.44	84.84	82.04	75.72	59.41
Proto-BERT-Remove (Snell et al., 2017)*	51.50	51.03	50.24	42.53	36.93	36.57	35.48	26.06
HATT-BERT (Gao et al., 2019a)*	89.52	88.84	86.71	74.51	83.46	82.34	80.08	69.65
TraNFS (Liang et al., 2022)*	89.33	88.27	86.52	74.19	83.01	82.30	79.77	69.09
CoT-ER (+ChatGPT) (Ma et al., 2023)*	90.21	85.30	79.05	70.45	82.36	75.49	66.84	52.53
MICRE (+Llama-2) (Li et al., 2024)*	89.86	85.05	80.11	71.90	81.74	75.26	67.34	54.76
Ours	**92.39**	**92.14**	**91.64**	**84.61**	**85.88**	**85.53**	**84.89**	**79.04**

Table 5: Accuracy (%) on the FewRel 2.0 test set. Results with * are obtained by our reproduction.

Model	20%	40%	60%
Ours	**95.56**	**94.06**	**90.08**
w/o. First iterative strategy	94.07	87.90	76.54
w/o. Second iterative strategy	92.84	88.02	83.03

Table 6: Accuracy of our model in correcting noisy labels in the support set. These experiments are conducted on the FewRel 1.0 validation set in the 5-way-5-shot setting.

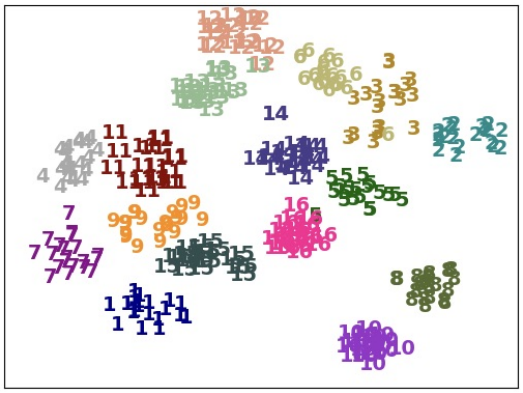


Figure 6: The t-SNE plots of instance representations produced by our GNN sublayer. These instances are from the validation set of FewRel 1.0. The number indicates the relation index.

removing either the first or second iteration strategy from our IC sublayer leads to a significant decline in model performance. These results suggest that both iterative strategies effectively improve the correction capability of our model.

D.2 Analysis for GNN sublayer

To further illustrate the effectiveness of our GNN sublayer, we visualize the distribution of instances in its embedding space. As shown in Figure 6, our GNN sublayer can effectively map instances into a better embedding space, where instances with different relations are clearly separated. As in our preliminary study, we use the Silhouette Coefficient to evaluate the discriminability among instances with different relations in the embedding space. In terms of this metric, our GNN sublayer achieves a higher score of 0.63 than Proto-BERT's 0.28 score, indicating that instances with different relations are better distinguished in the embedding space learned by the GNN sublayer. These results indicate that our feedback-based training strategy can indeed help the GNN sublayer learn a better embedding space, thus laying a solid foundation for the IC sublayer to correct the noisy labels of support instances.

D.3 Analysis for noise in the training set

In noisy few-shot learning, considering the absence of a strong correlation between the training and test sets due to their entirely distinct class spaces,

$\alpha\backslash\beta$	0%	20%	40%	60%
0%	**96.68**	**96.33**	**95.41**	**91.22**
5%	96.42	96.01	95.16	90.85
10%	96.10	95.95	94.74	90.79
15%	95.72	95.21	94.60	90.39
20%	95.14	94.67	94.08	89.68

Table 7: Accuracy (%) of our model on FewRel 1.0 validation set in the 5-way-5-shot setting. Here, α and β represent the noise rate in the training set and the support set of the test few-shot task, respectively.

previous studies (Mazumder et al., 2021; Liang et al., 2022) within the CV community usually presumed that relatively clean training sets could be found for model training. However, noisy labels are often inevitable in real-world scenarios. Here, therefore, we evaluate the robustness of our model to unknown noise in the training set. Specifically, we first randomly introduce some noisy labels into the entire training set, and then train our model on it. In particular, to avoid the influence of such unknown noise on our encoder, we use the training objective of HATT-BERT (Gao et al., 2019a) to pre-train our encoder. This training objective aims to mitigate the negative impact of unknown noise in the training set on model training by an attention mechanism. As shown in Table 7, our model is insensitive to this unknown noise and still exhibits strong robustness to noisy labels in support set. These results further demonstrate the practicality and robustness of our model.

Prompt	We now have two relations, each with three samples: The three samples of relation 1 are as follows: 1. The highest elevation in the [Innerste Uplands] is the 359m high Griesberg in the [Hildesheim Forest]. 2. The [Lamadaya] waterfalls is one of the best scenery in [Cal Madow]. 3. Contrary to some literature data, the [Vršač Mountains] are not part of the Carpathians but are a [Pannonian island mountain] according to their geotectonic position and geological structure. Three samples of relation 2 are as follows: 1. All the major dams controlling the [Nile] (the [Old Aswan Dam] and Esna, Nag Hammadi, Assiut, Edfeina and Zifta barrages) were constructed during this period. 2. The hill on the opposite side of the gorge to the west, the [Wittekindsberg], which is the eastern guardian of the [Wiehen Hills], defines the western side of this gorge. 3. The westernmost part of the Bavarian Prealps is formed by the [Ester Mountains] and its highest peak, the [Krottenkopf], which is also the highest summit in the Prealps. We use [ ] to mark the head entity and tail entity in the sample. Based on the options below, can you identify the above two relations? 1.located on terrain feature 2.located in the administrative territorial entity 3.mountain range Two of the samples are mislabeled. Can you identify them?
Chtat-GPT	Relation 1: located on terrain feature Relation 2: located in the administrative territorial entity Mislabelled samples: Sample 2 of relation 1 is mislabeled. It should be "mountain range". Sample 2 of relation 2 is mislabeled. It should be "located on terrain feature".
GPT-4	The two relations can be identified as follows: Relation 1: located on terrain feature Relation 2: mountain range The two mislabeled samples are: 1. "The [Lamadaya] waterfalls is one of the best scenery in [Cal Madow]." This should belong to relation 2 (mountain range) as it talks about a waterfall located in a mountain range. 2. "All the major dams controlling the [Nile] (the [Old Aswan Dam] and Esna, Nag Hammadi, Assiut, Edfeina and Zifta barrages) were constructed during this period." This doesn't seem to fit in either of the two relations as it talks about dams in a river, not a terrain feature or mountain range.
Ground-Truth	Relation 1: mountain range ; Relation 2: located on terrain feature The two mislabeled samples are: 1. Contrary to some literature data, the [Vršač Mountains] are not part of the Carpathians but are a [Pannonian island mountain] according to their geotectonic position and geological structure. 2. The hill on the opposite side of the gorge to the west, the [Wittekindsberg], which is the eastern guardian of the [Wiehen Hills], defines the western side of this gorge.

Figure 7: Results of Chat-GPT and GPT-4 identifying the two relations and mislabeled support instances in the case shown in Figure 1 of the paper. We observe that, even with sufficient hints, both Chat-GPT and GPT-4 are unable to accurately predict these two relations and mislabeled instances.

Case-1 ⟹ Relation 1/2/3: *mountain range*, *located on terrain feature*, *located in the administrative territorial entity*	
Relation 1	**1.** The highest elevation in the [**Innerste Uplands**] is the 359m high Griesberg in the [**Hildesheim Forest**]. **2.** The [**Lamadaya**] waterfalls is one of the best scenery in [**Cal Madow**]. **3.** Contrary to some literature data , the [**Vršač Mountains**] are not part of the Carpathians but are a **Pannonian island mountain** according to their geotectonic position and geological structure.
Relation 2	**1.** All the major dams controlling the [**Nile**] (the [**Old Aswan Dam**] and Esna, Nag Hammadi, Assiut, Edfeina and Zifta barrages) were constructed during this period. **2.** It was the Cougars ' first visit to and [**Idaho**]'s first loss in [**Neale Stadium**], which opened the previous year ; the Vandals had won the first five games played there. **3.** The westernmost part of the Bavarian Prealps is formed by the [**Ester Mountains**] and its highest peak , the [**Krottenkopf**], which is also the highest summit in the Prealps.
Relation 3	**1.** The Pithole Stone Arch Bridge is a masonry , deck arch bridge that spans [**Pithole Creek**] between Cornplanter and President Townships , Venango County in the U.S. state of [**Pennsylvania**]. **2.** The hill on the opposite side of the gorge to the west , the [**Wittekindsberg**], which is the eastern guardian of the [**Wiehen Hills**], defines the western side of this gorge. **3.** [**Madge Lake**] is the largest body of water in [**Saskatchewan**]'s Duck Mountain Provincial Park.
Case-2 ⟹ Relation 1/2/3: *biological process involves gene product*, *gene plays role in process*, *occurs in*	
Relation 1	**1.** Understanding how ubiquitin e2 and e3 ligases contribute to controlling [**mhc**] ii [**antigen presentation**] will shed light on the critical regulatory controls of this important pathway of immunity. **2.** Overexpression of [**pim2**] triggered anchorage-independent growth and resulted in increased bad [**phosphorylation**] and cell resistance to dna damaging agents. **3.** Our results showed that amp treatment activated [**autophagy**] in mda-mb-231 and mcf-7 breast cancer cells, as evidenced by the accumulation of autophagosomes, an increase of microtubule-associated protein 1 light chain 3 beta-2 ([**lc3b**]-ii) and the conversion of lc3b-i to lc3b-ii, the degradation of the selective autophagic target p62/sqstm1, and the formation of green fluorescent protein (gfp)-lc3 puncta.
Relation 2	**1.** The accumulation of these various organelles in the non-terminal [**axon**] in the absence of mechanical obstruction suggests a defect in [**axoplasmic transport**] which may result from an energy failure as suggested in toxic neuropathies. **2.** To define the [**signal transduction**] cascades downstream of sema4b for regulation of mmp9 expression , we inhibited [**pi3k**], erk/mapk , or jnk signaling pathway in sema4b knockout nsclc and found that only inhibition of pi3k signaling pathway significantly decreased mmp9 activation. **3.** The goal of this study was to assess in a hypertensive rat model if the early reperfusion with antihypertensive and pro-[**angiogenic**] chromogranin a-derived peptide, catestatin (cst: hcga352-372; cst-post), protects the heart via reperfusion-injury-salvage-kinases (risk)-pathway activation, limiting infarct-size and apoptosis, and promoting angiogenetic factors (e.g., hypoxia inducible factor, hif-1\u03b1, and endothelial nitric oxide synthase, [**enos**], expression).
Relation 3	**1.** The value of flow cytometry in the prediction of biologic behavior of [**congenital**] [**sct**] should be analyzed further. **2.** Pea enation mosaic virus (pemv) contains two adjacent 3‘cites in the center of its 703-nucleotide 3’ untranslated region (3‘utr), the ribosome-binding , kissing-loop t-shaped structure (kl-tss) and [**eukaryotic translation initiation factor 4e**]-binding panicum mosaic virus-like [**translation**] enhance (pte). **3.** The latest research has indicated that apart from the function of protecting [**cells**] from oxidation-induced stress damage , carnosine appears to be able to extend the lifespan of cultured cells , rejuvenate senescent cells , inhibit the toxic effects of amyloid peptide (a beta) , malondialdehyde , and hypochlorite to cells , inhibit glycosylation of proteins and protein-dna and protein-protein cross-linking , and maintain [**cellular homeostasis**].

Figure 8: Illustration of two suppor set with noisy lables (in 3-way-3-shot setting) from FewRel 1.0 and FewRel 2.0, respectively. The relation types are presented at the top of each case. The gray indicates the mislabeled instances. We use [**bold**] to indicate the head and tail entities in each instance. Please refer to the FewRel 1.0 and FewRel 2.0 datasets for more detailed information about these relations and instances