GLOBAL HARDEST EXAMPLE MINING WITH PROTOTYPE-BASED TRIPLET LOSS

Anonymous authors
Paper under double-blind review

ABSTRACT

Hard examples are the performance bottleneck of machine learning models, and therefore efficient identification and correct classification of them can significantly improve the model performance. However, most hard example mining schemes search for hard examples in randomly selected mini-batches at each epoch, which often result in local hardest examples and thus sub-optimal performances. Besides, the Triplet Loss is commonly adopted to explore the mined hard examples by pulling the hard positives close to and pushing the negatives away from the anchor. However, when the anchor in a triplet is an outlier at or close to the cluster boundary, the positive example will be pulled away from the centroid of the cluster, which would result in a incompact cluster, thus inferior performance. To address the above challenges, we propose a global hardest example mining with prototype-based Triplet Loss, which is composed of two major componets, namely Prototype-based Global Hardest Example Miner (GHEM) and Prototype-based Triplet Loss (pTriplet). First, a global hardest example miner (GHEM) is present to mine the global hardest classes on the prototype-based nearest neighbor graph of classes, and then the global hardest examples by searching for examples at the intersection between clusters. Second, a prototype-based Triplet Loss (pTriplet) is developed, which replaces the outlier anchor with a anchor-fused prototype to alleviate the influence of the outlier anchor and provides a normal anchor for Triplet Loss. Extensive experiments on typical Computer Vision (CV) and Natural Language Processing (NLP) tasks, namely person re-identification and few-shot relation extraction, demonstrated the effectiveness and generalizability of the proposed scheme, which consistently outperforms the-state-of-the-art models. We will publish all source codes of this work on GitHub for further research explorations.

1 INTRODUCTION

In machine learning, we generally refer to the examples in the training set that the model is prone to misclassify as hard examples (Sung, 1996; Dalal & Triggs, 2005), which is usually the bottleneck of model performance. To address this problem, hard example mining (HEM) algorithms specialize in mining hard examples that are most difficult to classify correctly from training examples and then let the model focus on learning them. These studies (Wen et al., 2016; Liu et al., 2017; Deng et al., 2019) shows that HEM can improve the performance of the model by refining the classification surface and has been applied to various tasks as person ReID Liao & Shao (2022), face recognition Smirnov et al. (2017; 2018); Qian et al. (2022) in CV, and few-shot relation extraction Ren et al. (2020), few-shot text classification Wei et al. (2021), text clustering Dor et al. (2018), sentence embedding Reimers & Gurevych (2019) in NLP.

Existing hard example mining schemes can be roughly classified into two categories, namely hard negative mining (HNM) schemes (Dollár et al., 2009; Felzenszwalb et al., 2010; Canévet & Fleuret, 2016; Jin et al., 2018) and online triplet mining schemes (Schroff et al., 2015; Shrivastava et al., 2016; Ge, 2018; Vasudeva et al., 2021), which will be introduced in detail below.

The HNM schemes generally use the updated model to mine informative hard examples from the whole training set and then use them to re-train the model. The research works of HEM in recent years usually first obtain hard classes by building the nearest neighbor graph on the training set (Suh

et al., 2019; Liao & Shao, 2022), and then randomly sample from hard classes as hard examples. Although these methods are effective, there are still the following problems: (1) the hard class may be the local hardest classes Sheng et al. (2020). (2) the hard examples randomly sampled from hard classes may be the local hardest examples. Specifically, an example is picked from the class cluster to represent the entire class, and then the nearest neighbor graph (NNG) is built by calculating the similarity between classes with them. However, when a class contains a large number of examples, the randomly selected examples may not necessarily represent the example distribution of the whole class, so obtaining the nearest neighbors may be the local hardest class. Second, the examples that are the most difficult to classify for the model are located at the intersection between clusters. In this scenario, they can be analogized to "support vectors" Li et al. (2018), and we name these examples as the global hardest examples. Hence, randomly sampling hard examples from clusters containing many examples may not necessarily be the global hardest examples. However, compared to instance-level, class prototypes can better represent the distribution of examples of the entire class, so the NNG built with prototypes can more accurately model the similarity between classes on the training set. Therefore, we urgently need to explore how to build the NNG based on the prototype to obtain the global hardest class and mine the global hardest examples at the intersection between clusters of the global hardest class.

The online triplet mining schemes generally pick hard examples from the mini-batch with a large gain to backpropagation at each iteration based on the loss value of the examples and then only use them to update the model. However, online triplet mining schemes pick hard examples in a mini-batch with limited information, so it is difficult to mine the hardest examples. But, while HEMs mine hard examples over the whole training set, so existing Triplet Loss (Schroff et al., 2015; Hermans et al., 2017) schemes work together with HNM, e.g., GS Liao & Shao (2022). Triplet Loss performs online hard example mining on mini-batch while updating the model at each step. It takes the positive examples farthest from the anchor as hard positive examples, both of which have the same identity. The negative examples, which are not the same ID as the anchor, closest to the anchor are regarded as hard negative examples. It minimizes the distance between anchors and hard positive examples and maximizes the distance between anchors and hard negative examples. But, when the anchor is an outlier (we call it the outlier anchor), the outlier anchor will lead to an excessively incompact cluster. Specifically, when the hard positive example is close to the outlier anchor, it may be far away from the cluster center and form the sub-cluster center centered on this outlier, eventually resulting in the excessively incompact cluster. When the hard negative examples are far away from outlier anchors rather than cluster centers, the above situation will also occur and cause the clusters of negative class to be excessively incompact. Finally, an excessively incompact cluster may lead to the intersection of clusters, which is difficult for the model to distinguish these clusters. Since the prototype is the cluster center, if outlier anchors fuse with the prototype to generate a new normal anchor near the cluster center, the above problem can be solved. However, in the training process, there are normal anchors and outlier anchors. So we need an adaptive anchor update strategy to identify outlier anchors and then correct them.

Based on the above analysis, we propose a global hardest example mining with prototype-based Triplet Loss. Firstly, the Prototype-based Global Hardest Example Miner (GHEM) finds global hardest classes by using the prototype to build the NNG, and its localization search strategy can mine the global hard example at the cluster intersection of global hardest classes. Secondly, the adaptive anchor correction strategy of Prototype-based Triplet Loss (pTriplet) fuses outlier anchors and prototypes to generate a normal anchor close to the cluster center, thus alleviating the excessively incompact cluster caused by the outlier anchor. We demonstrate the effectiveness and generalization of our schemes through experiments based on person re-identification (ReID) and few-shot relation extraction (FSRE) on typical tasks in CV and NLP, respectively.

2 Related Work

Hard example mining is one of the most important tasks in machine learning. Many efforts have been invested in hard example mining. Sung (1996) first proposed the Bootstrapping (hard negative mining) strategy, and Dalal & Triggs (2005) regarded false positives as hard examples, Shrivastava et al. (2016) formally proposed hard example mining. Existing hard example mining schemes are mainly divided into HNM Felzenszwalb et al. (2010); Jin et al. (2018); Dollár et al. (2009); Sun et al.

(2020); Qian et al. (2022) and Online Triplet Mining (Schroff et al., 2015; Hermans et al., 2017), as detailed below.

Hard Negative Mining. It freezes the deep neural network after a few iterations or each epoch and then uses the updated model to find hard examples on the whole training set. Bootstrapping Sung (1996) uses the updated model every few iterations to find hard examples on the training set when training deep neural networks, which will dramatically slow down the training progress. To speed up the training progress, Recent work is adapting training strategies. SmartMining Harwood et al. (2017) establishes an approximate NNG based on the instance-to-instance distance at the beginning of each training epoch. However, the build instance-level NNG is costly since all examples are considered. Wang et al. (2017) Randomly sample hard examples from hard classes obtained by K-means clustering. However, k-means is easy to converge to the local optimal solution Bottou & Bengio (1994), so the obtained hard class is not necessarily the global hardest class. Suh et al. (2019) get hard classes from NNG constructed based on instance-to-class distances, and then the hard example is obtained by random sampling from the hard class. Since this operation is performed at each iteration, it will bring huge computational costs. To alleviate these shortcomings, Liao & Shao (2022) randomly samples one example per class for NNG constructing based on instance-to-instance distance. Though effective, there are some shortcomings. Firstly, when a class contains tens of thousands of training examples, an example, which is randomly selected, may hard to represent this class. Besides, the selected hard examples, which are randomly sampled from the hard classes, may not locate at the intersection between clusters as mentioned in Introduction 1. Therefore, the hard examples obtained by such methods are the local hardest example. Different from the above methods, our proposed GHEM builds the NNG based on the prototype (i.e., class-to-class distance) to find the global hardest classes on the whole training dataset and then picks examples at the intersection between clusters, thereby efficiently mining the global hard example.

Online Triplet Mining. Schroff et al. (2015) proposed online triplet mining, which mines hard positives and hard negatives in each mini-batch based on the loss value to form a triplet (anchor, hard positive, hard negative) and then uses them to update the model. The remaining large number of triplets do not participate in the model update process, which greatly speeds up the convergence rate of the model. In order to solve the problem that Triplet Loss mining hard examples in mini-batch may cause local optimal solutions, Ge (2018) proposed to build a hierarchical class tree for all classes in the training set based on the similarity between classes. Cai et al. (2019) proposed Hard Exemplar Reweighting Triplet Loss, which weights the triplet according to their difficulty level. Vasudeva et al. (2021) proposed LoOp, which can be used to alleviate the biased embedding caused by Triplet Loss. To the best of our knowledge, existing research has not paid attention to the problem of the excessively incompact cluster caused by anchors being outliers. We propose pTriplet, which fuses outlier anchors and prototypes to generate a normal anchor close to the cluster center, thus alleviating this challenge

3 METHODOLOGY

Figure1 shows the architecture of our method, and we will detail our proposed methods, including global hardest example miner and prototype-based triplet loss, in this section. The first step is to mine the global hard example, as shown in Figure 1.A. First, use the model updated with the last epoch to get the prototype representation of each class on the training set, and then build the NNG based on the prototype to get the global hardest classes. Second, the global hardest examples are mined from the cluster intersection of these global hardest classes using our designed localization search strategy, which are used to construct the mini-batch. The second step is to correct the anchor, as shown in Figure 1.B. For the current epoch, the input mini-batch passes through the backbone to obtain the feature vector of the example, and then pTriplet corrects the outlier anchor to obtain the normal anchor. Then, pTriplet combines other losses (e.g., cross-entropy loss function) for back-propagation. After the current epoch training is completed, this process is repeated until the model converges.

Formally, given a training set $T = \{x_1, x_2, x_3, ..., x_n\}$, the training set has C categories, from which we pick the global hardest example set $H = \{h_1, h_2, h_3, ..., h_m\}$. GHEM mines the global hard example set H on the training set T before each epoch starts training and then uses them to construct the mini-batch for the model to train. In order to prevent the model from over-fitting

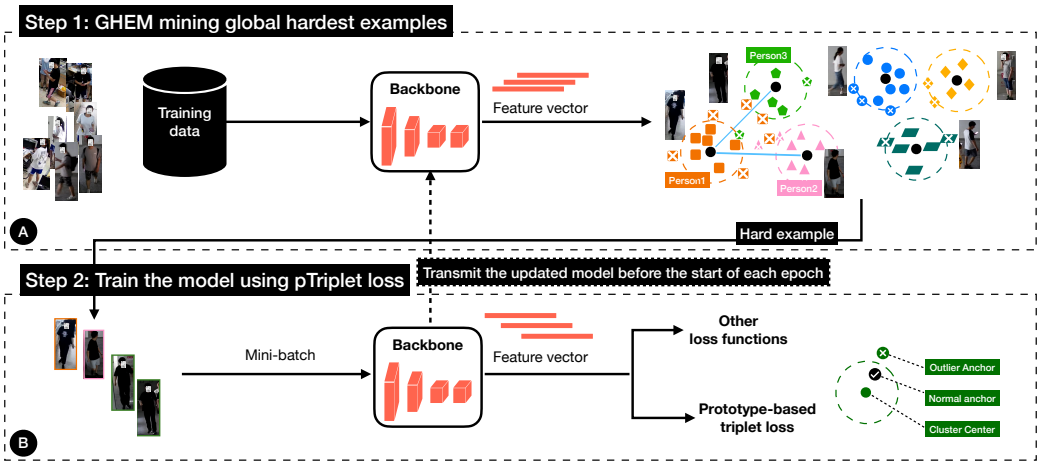

Figure 1: An overview of our scheme. Where the squares are normal examples, geometries with a cross are outliers, and different colors represent different classes. For the convenience of presentation, taking the person re-identification task as an example, we only construct the NNG of the class $Person1$ in Figure 1.A. The clusters connected by solid lines are nearest neighbor classes in the training datasets that are the most similar appearance to the target class $Person1$, and the remaining dissimilar classes that are not linked.

hard examples and to mine the latest hard examples with pTriplet as described above, the mini-batch contains randomly sampled examples in addition to H, which contains P categories and K examples in each category. In addition, given anchor A and prototype P, pTriplet fuses A and P to obtain a new normal anchor U. Our method consists of the following two parts:

1. **GHEM.** Given a training set T, the similarity between classes is modeled based on the NNG to obtain global hardest classes. Then, the location search strategy designed by us is used to mine the global hard example at the intersection between clusters of these global hardest classes.
2. **pTriplet.** Given the anchor A, the adaptive anchor correction strategy identifies the outlier anchor and then corrects it, thus generating a normal anchor U near the cluster center.

3.1 Global Hardest Example Miner

Because the examples that are the most difficult for the model to distinguish are located at the intersection between clusters as mentioned in Introduction 1. So our scheme aims to first find the corresponding classes of these clusters, that is, global hardest classes, and then mines the hard examples located at the intersection between clusters.

Constructing NNG for getting global hardest classes. As shown in Eq. 1, we obtain the prototype by averaging the example feature vectors in each class:

$$P_i = \frac{\sum_{j=1}^{k} f(x_{i,j})}{k}, \ i \in [1, C] \tag{1}$$

where f is backbone, i is the i-th class, and $f(x_{i,j})$ is the feature vector of the j-th example in the i-th class, P_i denotes the prototype of i-th class.

Then we build the NNG for all classes on the training set based on the prototypes, so then modeling the similar relationship between the classes. Firstly, for any class in the training set, we obtain its nearest neighbor class by comparing the cosine distance between this class and the rest of the classes. Since each class and its nearest neighbors are the closest in the feature vector space, we treat them as a set of global hardest classes.

$$\mathbb{H}_i = \{n_1, n_2, .., n_l\}, \ i \in [1, C] \tag{2}$$

where $\mathbb{H}_i$ represents the set of the global hardest classes consisting of the i-th class and its neighbors, l is the size of the $\mathbb{H}_i$, and n_i represents the i-th global hardest class.

Accordingly, NNG is constructed as shown in Eq 3.

$$\mathcal{G} = (\mathbb{V}, \mathbb{E}), \mathbb{V} = \{v_1, v_2, ..., v_C\}, \mathbb{E} = \{(v_i, v_j) \mid v_i \in \mathbb{H}(v_j)\}. \tag{3}$$

where $\mathbb{V}$ refers to all classes on the training set, and $\mathbb{E}$ refers to the set of edges.

Location search strategy gets global hard example. To obtain the global hardest example, we search among the clusters from the global hardest classes obtained above. For any two global hardest classes that are the nearest neighbor to each other, we obtain the vector M of their midpoints based on prototype P.

$$M = \frac{(P_u + P_v)}{2},\ u, v \in \mathbb{H}_i \tag{4}$$

where u and v denote global hardest class.

We take the sphere with m as the center and radius δ as the region where the clusters of the intersection between clusters of global hardest classes. Then we calculate the cosine distance between M and each example in global hardest classes, and then pick the example whose distance is less than δ as the global hard example.

$$\begin{aligned} &\mathbb{S} = \mathbb{S}_u \cup \mathbb{S}_v \\ &I = \{s \in \mathbb{S} \mid \cos(f(x_s), M) <= \delta\} \end{aligned} \tag{5}$$

where $\mathbb{S}_i$ represents the example set corresponding to the i-th class.

3.2 Prototype-based Triplet Loss

The purpose of pTrilet loss is to identify outlier anchors and then correct them. Inspired by the historical gradient used in stochastic optimization to reduce the gradient oscillation Kingma & Ba (2014), we use the exponentially weighted average formula to fuse the normal anchor in the current step with the prototype of the previous epoch to update the prototype to obtain a more robust prototype, and then fuse the outlier anchor and the updated prototype to generate a normal anchor near the cluster center.

Identify Outlier Anchors. We calculate the distance d between each example in the cluster and the prototype and take the example whose cosine distance d is greater than the threshold λ as outlier anchors. Otherwise, it is a normal example.

$$\begin{aligned} &d = 1 - \cos(p_i, x_j) \\ &\mathbb{O} = \{o \in \mathbb{S}_i \mid d > lambda\} \\ &\mathbb{N} = \{n \in \mathbb{S}_i \mid d <= lambda\} \end{aligned} \tag{6}$$

where $\mathbb{O}$ and $\mathbb{N}$ are outlier anchors and normal example sets, respectively.

Update the Prototype. At each step in the model learning process, we fuse the normal examples x^n and prototype P in the mini-batch using the exponentially weighted average formula to update the prototype vector.

$$P_i^t = \alpha \times P_i^{t-1} + (1 - \alpha) \times f(x_{i,j}^n),\ \alpha \in [0, 1] \tag{7}$$

where f is backbone, i refers to i-th class, j refers to j-th example, $f(x_{i,j}^a)$ is feature vector of the normal examples $x_{i,j}^n$. The P_i^t is the prototype of the ith class that has been iterated for t times, and α is the adjustment factor, which controls the proportion of P_i^{t-1} in the updated prototype P_i^{t-1}, so as to prevent the proportion of prototype P_i^{t-1} from being too large, which leads to over-compacting of clusters and over-fitting. The prototype obtained by GHEM before each epoch is used as the initial value of P for the current epoch.

Correct outlier anchors. The identified outlier anchors x^o and updated prototypes P^t are fused with the exponentially weighted average formula to generate a new normal anchor U close to the cluster center.

$$U = \beta \times P_i^t + (1 - \beta) \times f(x_{i,j}^o),\ \beta \in [0, 1] \tag{8}$$

4 Experiment

Due to space constraints, the details of the experiments (e.g., qualitative analysis) are included in the appendix. To verify the effectiveness and generalization of our scheme, we conduct experiments on typical CV and NLP tasks, namely person re-identification and few-shot relation extraction, the task description is as follows.

Table 1: Comparison of different hard example mining methods.

Method	Training set	MSMT17		Market1501		CUHK03-NP	
		R1	mAP	R1	mAP	R1	mAP
PK	Market1501	43.6	15.7	-	-	17.9	17.6
Cluster Wang et al. (2017)	Market1501	44.0	15.8	-	-	18.4	17.3
GS Liao & Shao (2022)	Market1501	45.9	17.2	-	-	19.1	18.1
Ours	Market1501	**48.4**	**18.3**	-	-	**19.2**	**18.5**
PK	MSMT17	-	-	75.9	45.3	16.4	17.0
Cluster Wang et al. (2017)	MSMT17	-	-	77.2	47.6	18.4	19.2
GS Liao & Shao (2022)	MSMT17	-	-	79.1	49.5	20.9	20.6
Ours	MSMT17	-	-	**79.2**	**50.1**	**21.1**	**21.3**
PK	MSMT17 (all)	-	-	79.5	52.3	22.8	23.3
Cluster Wang et al. (2017)	MSMT17 (all)	-	-	80.4	54.2	26.3	26.3
GS Liao & Shao (2022)	MSMT17 (all)	-	-	82.4	56.9	27.6	28.0
Ours	MSMT17 (all)	-	-	**82.6**	**58.1**	**30.2**	**30.3**
GS Liao & Shao (2022)	CUHK03-NP	46.9	15.4	68.2	37.3	-	-
Ours	CUHK03-NP	**48.4**	**16.3**	**69.8**	**38.2**	-	-

4.1 INTRODUCTION TO PERSON RE-IDENTIFICATION TASK

The ReID task is given a pedestrian image and then retrieve the pedestrian images across different cameras. ReID task is divided into the training phase and testing phase. In the training phase, we need to collect images of different pedestrians from the training set to form a mini-batch input into the network and let the model learn to distinguish different pedestrians. In the testing phase, the task of this phase is to retrieve images from the gallery with the same ID as the query and captured by different cameras. The model is responsible for extracting the feature vectors of the images in the query and gallery sets and then calculating the similarity between the images in the query and the gallery, finally taking the images with the highest similarity as the retrieval result. In the ReID task, we take pedestrians with similar appearances as hard examples. ReID faces many challenges such as illumination, orientation, occlusion, and especially pedestrians with similar appearanceYe et al. (2021). On the task of generalization person re-identification, (Deng et al., 2019; Liu et al., 2017; Wen et al., 2016) show that the model can refine the classification surface by learning hard examples, making the clusters more compact and more generalization. So we introduce the generalizable person re-identification task to verify the performance of different hard example mining schemes. Generalizable person re-identification generally refers to training on the source dataset and testing on the target dataset.

4.2 INTRODUCTION TO FEW-SHOT RELATION EXTRACTION TASK

FSRE is defined as when training examples are insufficient, given a sentence and an entity pair, the model needs to determine the relation that exists between entities, and represent it as an entity-relation triplet, i.e. (head entity, relation, tail entity) Han et al. (2020). The essence of FSRE is to do classification, that is, there are N categories of relations that need to be extracted, and the model needs to classify the sentences with the relation to be extracted into the one with the highest probability among the N categories. During the training phase, we mine sentences of different categories and semantically similar as hard examples. Test phase, given a sentence and entity pair, then the model predicts the relation that exists between the entities. This task tests whether the model can accurately model the nearest neighbor graph between classes in the complex semantic environment, and then mines the hard examples for the model to learn, finally improving the model's performance.

4.3 COMPARISON WITH SOTA HARD EXAMPLE MINING METHODS

Due to the small size of the CUHK03-NP and Market1501 test sets, the fluctuation of R1 by a few percentage points can not accurately reflect the performance of the model. However, the mAP metric is used to measure the overall retrieval performance, so we use mAP as the main metric on both

Table 2: Comparison with SOTA person re-identification methods. TL refers to the Triplet Loss. "+PK+TL" refers to using the PK sampler and Triplet Loss during training.

Backbone	Method	Size	MSMT17		Market1501		DukeMTMC	
			mAP	R1	mAP	R1	mAP	R1
CNN	CBN Zhuang et al. (2020)	256×128	42.9	72.8	77.3	91.3	67.3	82.5
	OSNet Zhou et al. (2019)	256×128	52.9	78.7	84.9	94.8	73.5	88.6
	MGN Wang et al. (2018)	384×128	52.1	76.9	86.9	95.7	78.4	88.7
	RGA-SC Zhang et al. (2020)	256×128	57.5	80.3	88.4	96.1	-	-
	SAN Jin et al. (2020b)	256×128	55.7	79.2	88.0	96.1	75.7	87.9
	SCSN Chen et al. (2020)	384×128	58.5	83.8	88.5	95.7	79.0	91.0
	ABDNet Chen et al. (2019)	384×128	60.8	82.3	88.3	95.6	78.6	89.0
	PGFA Miao et al. (2019)	256×128	-	-	76.8	91.2	65.5	82.6
	HOReID Wang et al. (2020)	256×128	-	-	84.9	94.2	75.6	86.9
	ISP Zhu et al. (2020)	256×128	-	-	88.6	95.3	80.0	89.6
DeiT-B/16	DeiT Touvron et al. (2021)+PK+TL	256×128	61.4	81.9	86.6	94.4	78.9	89.3
	TransReID He et al. (2021)+PK+TL	384×128	66.3	84.5	88.5	95.1	82.1	91.1
ViT-B/16	ViT Dosovitskiy et al. (2020)+PK+TL	256×128	61.0	81.8	86.8	94.7	79.3	88.8
	TransReID He et al. (2021)+PK+TL	384×128	69.4	86.2	89.5	95.2	82.6	90.7
ViT-B/16	TransReID+GS Liao & Shao (2022)+TL	384×128	70.0	86.9	89.6	**95.6**	83.4	91.2
	TransReID+Ours	384×128	**70.7**	**87.8**	**90.0**	95.5	**83.6**	**91.5**

datasets. By observing mAP, our method significantly outperforms all methods as shown in Table 1. Specifically, the PK sampler performs the worst because it constructs mini-batches by random sampling, which results in almost no hard examples to be mined. However, Cluster Wang et al. (2017) and GS Liao & Shao (2022) achieve better performance. The P classes sampled when they construct the mini-batch are not randomly sampled. Our scheme achieves respectively 2.5%/1.5% R1 and 1.1%/0.9% mAP gains over the SOTA GS on Market1501→MSMT17 /CUHK03-NP→MSMT17 (A→B means train on A's training set and test on B's test set) on the MSMT17 data set with a huge test set. Our scheme outperforms all methods on mAP, and the possible reasons are as follows: since mAP reflects the overall retrieval effect, the increase in mAP indicates that more clusters are accurately classified by the classification surface. At this time, for clusters, there is a discriminative margin between different clusters, and the clusters are more compact, which indicates that our proposed GHEM can mine the global hard example at the intersection between clusters, and the model distinguishes them correctly. At the same time, it reduces the intersection between clusters. Meanwhile, pTriplet corrects outliers to make clusters more compact and the classification surface easier to classify. For further quantitative analysis of the superiority of our proposed scheme, readers can refer to cases of hard example mining and distribution of the intra-class similarity provided by section C.1 and C.2 respectively in the appendix.

4.4 PERFORMANCE COMPARISON ON PERSON RE-IDENTIFICATION

4.4.1 COMPARISON WITH SOTA MODELS ON PERSON RE-IDENTIFICATION TASKS

Table 2 shows the performance of models on the test set of different datasets. It can be observed that our proposed scheme achieves the best performance compared to other schemes. Specifically, a performance gain of 1.3% R1 and 1.6% mAP is registered as compared to TransReID on the msmt17 dataset, respectively. And compared with GS, our method gains 0.7% R1 and 0.9% mAP on the msmt17 dataset, respectively. There may be two reasons why our method outperforms GS. First, our method builds the NNG based on the prototype, while GS randomly selects an example from the class and then uses this example to represent the class to build the NNG. The prototype is more representative of the example distribution of the class than the selected example, so the obtained nearest neighbors are the global hardest classes rather than the local hardest classes. Second, GHEM focuses on the global hardest examples at the intersection between clusters ignored by GS. Relative qualitative retrieving results in section C.3 of the appendix.

4.4.2 PERFORMANCE COMPARISON OF THE MODEL ON THE GENERALIZATION PERSON RE-IDENTIFICATION TASK OF BACKBONE USING RESNET

Table 3 shows the SOTA direct cross-dataset evaluation results, where each group is the model trained on the training set of the source dataset and then tested on the test set of the target dataset. It can be observed that our proposed scheme achieves the best performance compared to other schemes. For example, with MSMT17(all)→CUHK03-NP, our proposed scheme achieves 2.6% R1 and 2.3% mAP gains over the GS, respectively. The possible reason why our strategy outperforms GS is that our scheme fully takes into account global hardest examples at the intersection between clusters. On the contrary, the examples are randomly sampled by GS from each cluster as the hard examples are not necessarily the global hardest examples. The pTriplet corrects outlier anchors when calculating loss, which will make the cluster converge more compactly, thus making the classification surface of the model more distinguishable, and more generalizable on the generalizable person re-identification task.

Table 3: Direct cross-dataset evaluation results. MSMT17 (all) means that the model uses both the test set and the training set for training. M^3L selects three datasets from CUHK03, Market1501, DukeMTMC-reID, and MSMT17 for training and the remaining one for testing. TL refers to the Triplet Loss. "+PK+TL" refers to using the PK sampler and Triplet Loss during training.

Method	Training set	CUHK03-NP		Market1501		MSMT17	
		R1	mAP	R1	mAP	R1	mAP
M^3L Zhao et al. (2021)	Multi	33.1	32.1	75.9	50.2	36.9	14.7
MGN Wang et al. (2018)	Market1501	8.5	7.4	-	-	-	-
MuDeep Qian et al. (2019)	Market1501	10.3	9.1	-	-	-	-
QAConv Liao & Shao (2020)	Market1501	9.9	8.6	-	-	22.6	7.0
OSNet-AIN Zhou et al. (2021)	Market1501	-	-	-	-	23.5	8.2
CBN Zhuang et al. (2020)	Market1501	-	-	-	-	25.3	9.5
QAConv + GS Liao & Shao (2022) + TL	Market1501	19.1	18.1	-	-	45.9	17.2
QAConv + Ours	Market1501	**19.2**	**18.5**	-	-	**48.4**	**18.3**
PCB Sun et al. (2018)	MSMT17	-	-	52.7	26.7	-	-
MGN Wang et al. (2018)	MSMT17	-	-	48.7	25.1	-	-
ADIN Yuan et al. (2020)	MSMT17	-	-	59.1	30.3	-	-
SNR Jin et al. (2020a)	MSMT17	-	-	70.1	41.4	-	-
CBN Zhuang et al. (2020)	MSMT17	-	-	73.7	45.0	-	-
QAConv + GS Liao & Shao (2022)	MSMT17	20.9	20.6	79.1	49.5	-	-
QAConv + Ours	MSMT17	**21.1**	**21.3**	**79.2**	**50.1**	-	-
OSNet-IBN Zhou et al. (2019)	MSMT17 (all)	-	-	66.5	37.2	-	-
OSNet-AIN Zhou et al. (2021)	MSMT17 (all)	-	-	70.1	43.3	-	-
QAConv Liao & Shao (2020)+ PK + TL	MSMT17 (all)	25.3	22.6	72.6	43.1	-	-
QAConv + GS Liao & Shao (2022) + TL	MSMT17 (all)	27.6	28.0	82.4	56.9	-	-
QAConv + Ours	MSMT17 (all)	**30.2**	**30.3**	**82.6**	**58.1**	-	-
QAConv + GS Liao & Shao (2022) + TL	CUHK03-NP	-	-	68.2	37.3	46.9	15.4
QAConv + Ours	CUHK03-NP	-	-	**69.8**	**38.2**	**48.4**	**16.3**

4.5 COMPARISON WITH SOTA MODELS ON FEW-SHOT RELATION EXTRACTION TASKS

Table 4 shows the performance of models using different hard example mining schemes on the test set of FewRel Han et al. (2018) dataset. The first and second groups of schemes use PK sampler, the third group of ConceptFERE uses SOTA hard example mining strategy GS and Triplet Loss, and the fourth group of ConceptFERE uses our proposed scheme. It should be noted that, due to the insufficient computing power of our GPU, the performance of the proposed scheme is tested only under 5 ways 1 shot and 10 ways 1 shot scenarios. It can be observed from Table 4 that compared with all the comparison schemes, our proposed scheme achieves the best performance. More specifically, our proposed scheme achieves gains of 1.03% and 3.13% compared with ConceptFERE, under the scenarios of 5-w-1-s and 10-w-1-s, the best model in the second group, respectively. The possible reason is that PK Sampler randomly selects classes and examples, and it is almost impossible to

Table 4: Accuracies (%) of different models on the test set. TL refers to the Triplet Loss. Due to the randomness of the FSRE experiment, we take the average of ten experimental results as the final result. "+PK+TL" refers to using the PK sampler and Triplet Loss during training.

Backbone	Sampler	Model	5-w-1-s	5-w-5-s	10-w-1-s	10-w-5-s
CNN	PK	Gnearest neighbor Satorras & Estrach (2018)	67.30	78.84	54.10	62.89
		SNAIL Mishra et al. (2018)	71.13	80.04	50.61	66.68
		Proto Snell et al. (2017)	74.29	85.18	61.15	74.41
		HATT-Proto Gao et al. (2019a)	74.84	85.81	62.05	75.25
		MLMAN Ye & Ling (2019)	78.21	88.01	65.70	78.35
BERT	PK	Bert-PAIR Gao et al. (2019b)	82.57	88.47	73.37	81.10
		TD-Proto Yang et al. (2020)	84.76	92.38	74.32	85.92
		ConceptFERE(Simple) Yang et al. (2021)	84.28	90.34	74.00	81.82
		ConceptFERE Yang et al. (2021)	89.21	–	75.72	–
BERT	GS	ConceptFERE+GS Liao & Shao (2022)+TL	89.37	–	75.75	–
BERT	Ours	ConceptFERE+Ours	**90.24**	–	**78.85**	–

mine hard examples. More importantly, our proposed scheme achieves 0.87% and 3.1% gains over the GS under the scenarios of 5-w-1-s and 10-w-1-s, respectively. This may be because GS, in the natural language scene with thousands of examples in each category and complicated semantics, can't accurately model the relationship between classes by building an NNG by randomly selecting an example as the representative of the class, so it's difficult to mine global hardest classes. And it also ignores the global hard example at the intersection between clusters. Theoretically, 1-shot relation extraction is more difficult than 5-shot relation extraction, and the experimental results in the 1-shot scenario have illustrated the effectiveness and superiority of our scheme. We believe that our strategy can achieve better performance under the other two scenarios.

4.6 Ablation Study

In this section, we verify the effectiveness of the proposed GHEM and pTriplet presented in 3.2 and 3.3, respectively. As shown in Table 5 and Tabel 6, without GHEM and pTriplet, the performance of TransReID and ConceptFERE drops sharply. This proves that the proposed GHEM can effectively mine the global hard example, and the pTriplet corrects outlier anchors for the Triplet Loss to work efficiently.

Table 5: Results of ablation study on MSMT17 dataset of person ReID task.

Scheme	R1	mAP
TransReID + Ours	87.8	70.7
w/o GHEM	87.2	70.2
w/o pTriplet	87.0	70.1
w/o GHEM & pTriplet	86.2	69.4

Table 6: Results of ablation study on FSRE task.

Scheme	10-w-1-s
ConceptFERE + Ours	78.85
w/o GHEM	76.81
w/o pTriplet	77.12
w/o GHEM & pTriplet	75.72

5 Conclusion

In this paper, we studied the problem of hard example mining, which is essential for deep learning algorithms. Our designed GHEM can accurately mine the global hard example by the prototype-based NNG and location search strategy. Our designed pTriplet corrects the outlier anchor for Triplet Loss so that it can produce a more compact cluster. And the experimental results demonstrate the effectiveness and generalization of our scheme. In the future work, we will pay attention to the outliers near the classification surface, and quickly build the nearest neighbor graph with the outliers as nodes, and do more fine-grained mining of hard examples.

REFERENCES

Leon Bottou and Yoshua Bengio. Convergence properties of the k-means algorithms. *Advances in neural information processing systems*, 7, 1994.

Sudong Cai, Yulan Guo, Salman Khan, Jiwei Hu, and Gongjian Wen. Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8391–8400, 2019.

Olivier Canévet and François Fleuret. Large scale hard sample mining with monte carlo tree search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5128–5137, 2016.

Tianlong Chen, Shaojin Ding, Jingyi Xie, Ye Yuan, Wuyang Chen, Yang Yang, Zhou Ren, and Zhangyang Wang. Abd-net: Attentive but diverse person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8351–8361, 2019.

Xuesong Chen, Canmiao Fu, Yong Zhao, Feng Zheng, Jingkuan Song, Rongrong Ji, and Yi Yang. Salience-guided cascaded suppression network for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3300–3310, 2020.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:886–893 vol. 1, 2005.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral channel features. 2009.

Liat Ein Dor, Yosi Mass, Alon Halfon, Elad Venezian, Ilya Shnayderman, Ranit Aharonov, and Noam Slonim. Learning thematic similarity metric from article sections using triplet networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 49–54, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6407–6414, 2019a.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fewrel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6250–6255, 2019b.

Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 269–285, 2018.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147*, 2018.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. More data, more relations, more context and more openness: A review and outlook for relation extraction. *CoRR*, abs/2004.03186, 2020. URL `https://arxiv.org/abs/2004.03186`.

Ben Harwood, Vijay Kumar BG, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2821–2829, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer ViCVPR'21sion*, pp. 15013–15022, 2021.

Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

SouYoung Jin, Aruni RoyChowdhury, Huaizu Jiang, Ashish Singh, Aditya Prasad, Deep Chakraborty, and Erik Learned-Miller. Unsupervised hard example mining from videos for improved object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 307–324, 2018.

Xin Jin, Cuiling Lan, Wenjun Zeng, Zhibo Chen, and Li Zhang. Style normalization and restitution for generalizable person re-identification. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3143–3152, 2020a.

Xin Jin, Cuiling Lan, Wenjun Zeng, Guoqiang Wei, and Zhibo Chen. Semantics-aligned representation learning for person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11173–11180, 2020b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kai Li, Zhengming Ding, Kunpeng Li, Yulun Zhang, and Yun Fu. Support neighbor loss for person re-identification. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 1492–1500, 2018.

Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 152–159, 2014.

Shengcai Liao and Ling Shao. Interpretable and generalizable person re-identification with query-adaptive convolution and temporal lifting. In *European Conference on Computer Vision*, pp. 456–474. Springer, 2020.

Shengcai Liao and Ling Shao. Graph Sampling Based Deep Metric Learning for Generalizable Person Re-Identification. June 2022.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.

Jiaxu Miao, Yu Wu, Ping Liu, Yuhang Ding, and Yi Yang. Pose-guided feature alignment for occluded person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 542–551, 2019.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Jianjun Qian, Shumin Zhu, Chaoyu Zhao, Jian Yang, and Wai Keung Wong. Otface: Hard samples guided optimal transport loss for deep face representation. *arXiv preprint arXiv:2203.14461*, 2022.

Xuelin Qian, Yanwei Fu, Tao Xiang, Yu-Gang Jiang, and Xiangyang Xue. Leader-based multi-scale attention deep architecture for person re-identification. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):371–385, 2019.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Haopeng Ren, Yi Cai, Xiaofeng Chen, Guohua Wang, and Qing Li. A two-phase prototypical network model for incremental few-shot relation classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1618–1629, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.142. URL https://aclanthology.org/2020.coling-main.142.

Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, pp. 17–35. Springer, 2016.

Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*, 2018.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Hao Sheng, Yanwei Zheng, Wei Ke, Dongxiao Yu, Xiuzhen Cheng, Weifeng Lyu, and Zhang Xiong. Mining hard samples globally and efficiently for person reidentification. *IEEE Internet of Things Journal*, 7(10):9611–9622, 2020.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.

Evgeny Smirnov, Aleksandr Melnikov, Sergey Novoselov, Eugene Luckyanets, and Galina Lavrentyeva. Doppelganger mining for face representation learning. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1916–1923, 2017.

Evgeny Smirnov, Aleksandr Melnikov, Andrei Oleinik, Elizaveta Ivanova, Ilya Kalinovskiy, and Eugene Luckyanets. Hard example mining with auxiliary embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 37–46, 2018.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pp. 4077–4087, 2017.

Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. Stochastic class-based hard example mining for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7251–7259, 2019.

Qianru Sun, Yaoyao Liu, Zhaozheng Chen, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning through hard tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European conference on computer vision (ECCV)*, pp. 480–496, 2018.

Kah-Kay Sung. Learning and example selection for object and pattern detection. 1996.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Bhavya Vasudeva, Puneesh Deora, Saumik Bhattacharya, Umapada Pal, and Sukalpa Chanda. Loop: Looking for optimal hard negative embeddings for deep metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10634–10643, 2021.

Chong Wang, Xue Zhang, and Xipeng Lan. How to train triplet networks with 100k identities? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1907–1915, 2017.

Guan'an Wang, Shuo Yang, Huanyu Liu, Zhicheng Wang, Yang Yang, Shuliang Wang, Gang Yu, Erjin Zhou, and Jian Sun. High-order information matters: Learning relation and topology for occluded person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6449–6458, 2020.

Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 274–282, 2018.

Jason Wei, Chengyu Huang, Soroush Vosoughi, Yu Cheng, and Shiqi Xu. Few-shot text classification with triplet networks, data augmentation, and curriculum learning. *arXiv preprint arXiv:2103.07552*, 2021.

Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 79–88, 2018.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pp. 499–515. Springer, 2016.

Kaijia Yang, Nantao Zheng, Xinyu Dai, Liang He, Shujian Huang, and Jiajun Chen. Enhance prototypical network with text descriptions for few-shot relation classification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2273–2276, 2020.

Shan Yang, Yongfei Zhang, Guanglin Niu, Qinghua Zhao, and Shiliang Pu. Entity concept-enhanced few-shot relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 987–991, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.124. URL https://aclanthology.org/2021.acl-short.124.

Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Zhi-Xiu Ye and Zhen-Hua Ling. Multi-level matching and aggregation network for few-shot relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2872–2881, 2019.

Ye Yuan, Wuyang Chen, Tianlong Chen, Yang Yang, Zhou Ren, Zhangyang Wang, and Gang Hua. Calibrated domain-invariant learning for highly generalizable large scale re-identification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3589–3598, 2020.

Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Xin Jin, and Zhibo Chen. Relation-aware global attention for person re-identification. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 3186–3195, 2020.

Yuyang Zhao, Zhun Zhong, Fengxiang Yang, Zhiming Luo, Yaojin Lin, Shaozi Li, and Nicu Sebe. Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6277–6286, 2021.

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pp. 1116–1124, 2015.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3702–3712, 2019.

Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Learning generalisable omni-scale representations for person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Kuan Zhu, Haiyun Guo, Zhiwei Liu, Ming Tang, and Jinqiao Wang. Identity-guided human semantic parsing for person re-identification. In *European Conference on Computer Vision*, pp. 346–363. Springer, 2020.

Zijie Zhuang, Longhui Wei, Lingxi Xie, Tianyu Zhang, Hengheng Zhang, Haozhe Wu, Haizhou Ai, and Qi Tian. Rethinking the distribution gap of person re-identification with camera-based batch normalization. In *European Conference on Computer Vision*, pp. 140–157. Springer, 2020.

A RANDOMNESS

We choose an arbitrary random seed and conduct experiments on the ReID task with this seed. Due to the randomness of the FSRE experiment, we take the average of ten experimental results as the final experimental result.

B EXPERIMENT

B.1 PERSON RE-IDENTIFICATION

B.1.1 DATASETS, EVALUATION METRICS, BASELINE, COMPARABLE SCHEME, AND EXPERIMENTAL SETTINGS

Datasets: On person re-identification and generalizable person re-identification tasks, we use four datasets to validate our proposed scheme, namely: Market1501 Zheng et al. (2015), DukeMTMC-reID Ristani et al. (2016), MSMT17 Wei et al. (2018), CUHK Li et al. (2014). The details of the datasets are summarized in Table 7. It should be pointed out that on the generalizable person re-identification task, cross-dataset evaluation is performed by training the model on the training set of the source dataset, and then evaluating it on the test set of the target dataset.

Table 7: ID_q, ID_g, ID_t represent the number of IDs (pedestrians) in the query set, gallery set, and training set, respectively. IMG_q, IMG_g, IMG_t represent the number of images in the query set, gallery set, and training set respectively. CAM_n represents the number of cameras in the dataset.

Dataset	ID_q	ID_g	ID_t	IMG_q	IMG_g	IMG_t	CAM_n
MSMT17	3060	3060	1041	11659	82161	32621	15
Market1501	750	750	751	3368	19732	12936	6
DukeMTMC-reID	702	702	702	2228	17661	16522	8
CUHK03-NP	700	700	767	1400	5332	7365	2

Evaluation Metrics: Rank-1 (R1) and mean average precision (mAP) are used as evaluation metrics. R1 is the correct rate of the first one in the retrieve results. The mAP refers to the correct rate of the overall ranking of the retrieved results. All experimental evaluations follow the single-query evaluation protocolLi et al. (2014).

Baseline: We use the commonly used PK sampler as the baseline of hard negative mining, which randomly picks P pedestrians from the training set, and each pedestrian randomly picks K images to construct a mini-batch. We use Triplet Loss as the baseline for online hard example mining Schroff et al. (2015).

Comparable Scheme: For the comparison of the hard example mining schemes, we use Cluster Wang et al. (2017), SOTA GS Liao & Shao (2022). It should be noted that since the Cluster Wang et al. (2017) is not open source code, we cite its experimental results only on the generalized person re-identification task inLiao & Shao (2022).

Experimental Settings: Our scheme is based on the official PyTorch code of TransReID[1] He et al. (2021) and GS[2] Liao & Shao (2020). In the person re-identification task, the backbone uses ViT Dosovitskiy et al. (2020), and the data augmentation strategy used is random horizontal flipping, padding, random cropping, and random erasing Zhong et al. (2020). On the generalizable person re-identification task, the backbone is ResNet50 He et al. (2016), with IBN-b layers appended. And the data augmentation strategies include random cropping, flipping, occlusion, and color jittering. The input image is resized to 384×128, and the loss function is Triplet Loss or cross-entropy loss. The rest of the hyperparameters, e.g., batch size, and the learning rate of the optimizer, follows the settings of TransReID He et al. (2021) and GS Liao & Shao (2020). Some columns in the table are empty, indicating that the method has no test results on the corresponding dataset or the generalized person re-identification task. The source dataset is not tested.

[1] https://github.com/damo-cv/TransReID

[2] https://github.com/ShengcaiLiao/QAConv

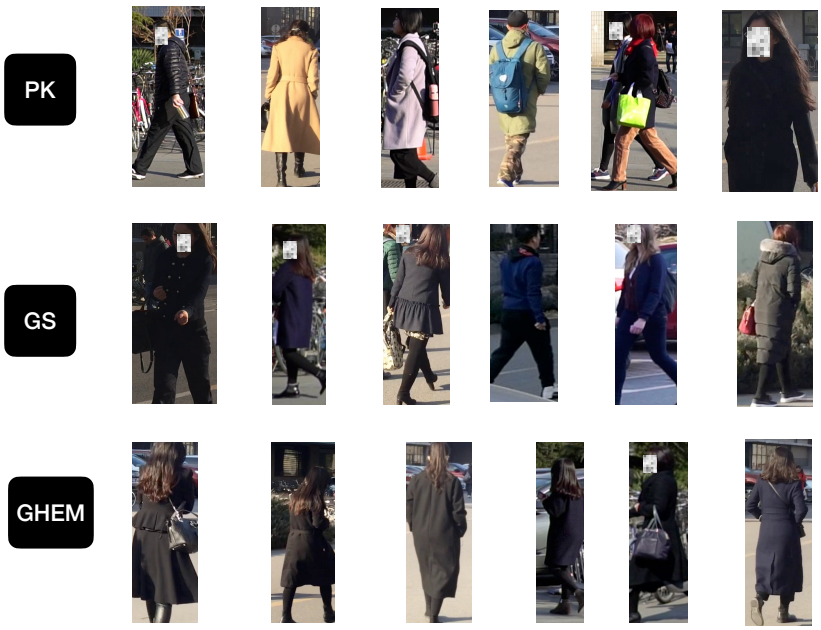

Figure 2: Due to the limited space, we only show some people in the MSMT17 dataset. PK randomly selects six people, the first image of GS and GHEM is the target class, and the remaining five images correspond to the five nearest neighbor classes.

B.2 Few-shot Relation Extraction

B.2.1 Datasets, Evaluation Metrics, Baseline and Comparable Scheme, Experimental Settings

Dataset: In order to verify our proposed scheme, we use the most commonly used few-shot relation extraction dataset FewRel Han et al. (2018), which contains 100 relations and 70,000 instances extracted from Wikipedia, with 20 relations in the unpublished test set. So we follow previous work Yang et al. (2021) to re-split the published 80 relations into 50, 14, and 16 for training, validation, and testing, respectively.

Evaluation: N-way-K-shot (N-w-K-s) is commonly used to simulate the distribution of few-shot relation extraction in different situations, where N and K denote the number of classes and examples from each class, respectively. In N-w-K-s scenario, accuracy is used as the performance metric.

Baseline: We use commonly used PK sampler as the baseline of hard negative mining and use Triplet Loss as the baseline for online hard example mining Schroff et al. (2015).

Comparable Scheme: To the best of our knowledge, there is no dedicated hard example mining algorithm in few-shot relation extraction, and almost all hard example mining algorithms used in NLP come from CV. So we choose the GS in CV as the comparison scheme. We choose excellent ConceptFERE Yang et al. (2021) as the comparable model.
Experimental Settings The BERTDevlin et al. (2018) parameters are initialized by bert-base-uncased, and the hidden size is 768. Hyperparameters such as learning rate follow the settings in ConceptFERE Yang et al. (2021).

B.2.2 Model Training Details

When mining hard examples in the data sampling stage, taking 5-w-1-s as an example, our scheme will randomly sample a target class. And then, find the top-4 nearest neighbor in the training set as the global hardest class. Each of sampled five classes finds its global hardest examples, and then

randomly assigns hard examples to the query set and support set. Our proposed scheme and GS are implemented on ConceptFERE[3].

C Qualitative Analysis

C.1 Hard Example Mining Cases

In order to intuitively observe the ability of our proposed GHEM to mine hard examples, We randomly select a mini-batch from mini-batches constructed by PK Sampler, GS, and GHEM, respectively, as shown in Figure 2. By observing the case in Figure 2, GHEM has a stronger ability to mine hard examples.

C.2 Evidence for clusters becoming compact

To verify that our proposed pTriplet can generate more compact clusters, we show in Table 8 the intra-class variance distributions produced when using Triplet Loss and pTriplet, respectively. The smaller the variance, the higher the similarity between examples in a class, the closer they are in space, and the more compact the cluster. On the ReID task, and the variance distribution of the intra-class similarity is shown in Table 8. The intra-class variance is the variance of the cosine distance between any two examples within the class. By observing Table 8, the clusters produced by pTriplet are more compact.

Table 8: The variance distribution of the intra-class similarity on MSMT17.

Model	Variance	Sampler	Online Hard Example Loss
TransReID	45.2	PK	Triplet Loss
	32.9	PK	pTriplet

C.3 Compare predict results

In order to intuitively observe the impact of different hard example mining algorithms on the model to classify hard examples, we take the person ReID task as an example and randomly select a group from the retrieve results to show in Figure 3. By observing the case in Figure 3, it can be found that models trained with our scheme are more capable of discriminating hard examples.

[3]https://github.com/LittleGuoKe/ConceptFERE

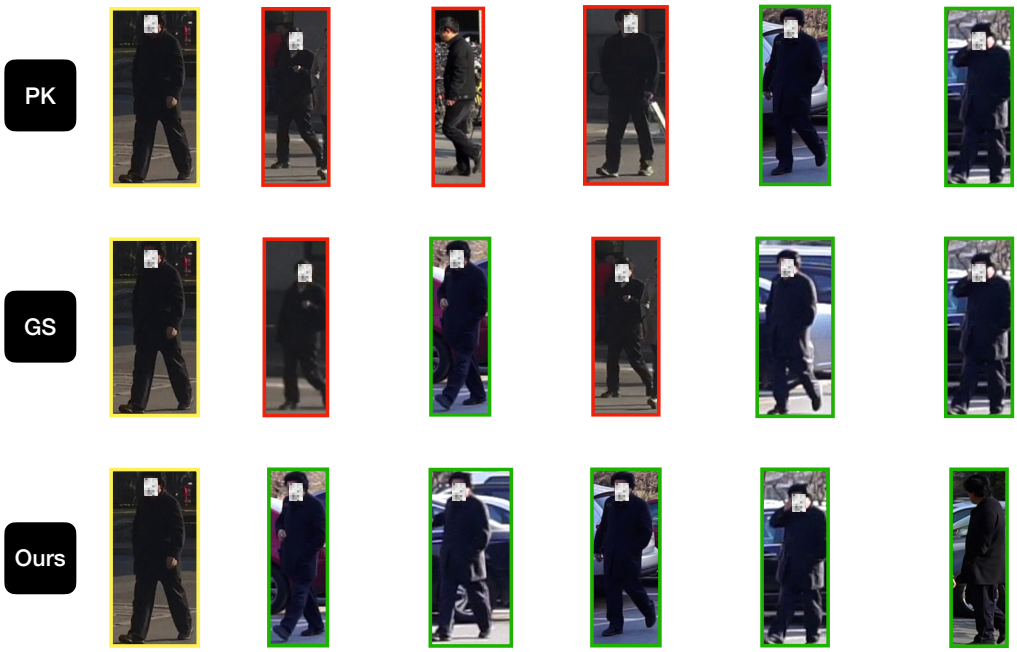

Figure 3: Top-5 of the retrieval results of the models trained with PK, GS, and ours, respectively. The pictures with a yellow box, green box, and red box are query, positive example, negative example, respectively